Abstract

KELKAR, INDRANEEL BALKRISHNA. Tradeoffs Involved in Design of SRAMs. (Under the direction of Dr. Paul D Franzon).

This thesis explores the tradeoffs that are involved in the design of SRAMs. The major components of an SRAM such as the row decoders, the memory cells and the sense amplifiers have been studied in detail. The circuit techniques used to reduce the power dissipation and delay of these components have been explored and the tradeoffs have been explained.

The key to low power operation in the SRAM data path is to reduce the signal swings on the high capacitance nodes like the bitlines and the data lines. A unique resetting scheme for the row decoders has been proposed. Using this technique the word line pulse width can be minimized and the signal swings on the bitlines is reduced.

Finally a 4Kb prototype SRAM has been designed and verified. This design incorporates some of the circuit techniques used to reduce power dissipation and delay. Experimental data has been provided which shows the effectiveness of using the resetting scheme for the row decoders. The design was simulated at a clock speed of 500Mhz. The read access time was found to be 0.83ns while the write access time was found to be 0.62ns. The total power dissipation was 26.3mW.

Tradeoffs Involved in Design of SRAMs

by

Indraneel B Kelkar

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Master of Science in

Computer Engineering

Raleigh

2005

Approved By:

Dr. Kevin G Gard	Dr. W. Rhett Davis

Dr. Paul D Franzon
Chair of Advisory Committee

Dedication

To the people who make this all worth while:

To Mom and Dad

To my uncle (Ram kaka).

Without your constant support and motivation this would not have been possible.

Biography

Indraneel Kelkar was born in Pune, India in 1981. He traveled across most of India during his childhood, finally settling down in Pune from his 7th grade onwards. He completed his Bachelors degree in Electronics from Vishwakarma Institute of Technology, (University of Pune) in 2003. He joined NC State for his Master's program in Fall 2003. It was here that his interest in VLSI circuit design grew and after a 7 month co-op at Eaton Corporation he started work under the guidance of Dr. Franzon.

Acknowledgements

I would like to extend a sincere thank you to my academic advisor, Dr. Paul Franzon, for his guidance and support. I would also like to thank my committee members Dr. Gard and Dr. Davis.

A big thanks to all my friends and in particular to Renuka. Thanks a lot for all the support and constant motivation.

Contents

List of Tables viii

List of Figures ix

1 Introduction 1

2 Overview of SRAM 4
2.1 Operation of a basic Random Access Memory 4
2.1.1 SRAM architectures 4
2.1.2 Circuit Techniques in SRAM design 8
2.1.2.1 Memory Cell Types 8
2.1.2.2 Row Decoders 9
2.1.2.3 Sense Amplifiers 11
2.2 Conclusion 13

3 Fast Low Power SRAM Decoder and Data Path Design 14
3.1 Design of Fast Low Power Decoders 14
3.1.1 Sizing and Buffering Considerations 15
3.1.2 Circuit techniques involved in decoder design 16
3.1.2.1 Reducing the Logical Effort 16
3.1.2.2 Different decoder design schemes 19
3.2 Memory Cell Types 21
3.2.1 Mechanism of Cell Instability 22
3.3 Sense Amplifiers 24
3.3.1 Methods of sensing 24
3.3.2 Voltage Sense Amplifiers 25
3.3.2.1 Static Voltage Sense Amplifiers 25
3.3.2.2 Dynamic Voltage Sense Amplifier 26
3.3.3 Current sense amplifiers 27
3.3.3.1 Basic working principle of current sense amplifiers . . 27
3.3.3.2 Type A - Cascode and Diode Connection 28
3.3.3.3 Type B - Regulated Cascodes 29

3.3.3.4 Type C - Negative Impedance Converter 29
3.3.3.5 Type D - Wilson Type 31
3.4 Circuit Techniques to generate Sense Enable Signal 32
3.4.1 Replica delay based on capacitance ratioing 34
3.4.2 Replica delay element based on cell current ratioing 36
3.5 Compensation schemes for Bitline Multiplexer 37
3.5.1 General considerations for design of multiplexer 38
3.5.2 Techniques for Multiplexer Compensation 39
3.6 Low Power Writes 40

4 Design of 4K SRAM 43
4.1 Design Architecture 43
4.2 Decoder Design 45
4.2.1 Decoder Architecture 46
4.2.2 Design of SCL Predecoder Circuits 48
4.2.3 Design of Reset Circuit 50
4.2.4 Design of Local Word Line Drivers 51
4.2.5 Conclusion 52
4.3 Memory Cell Design 53
4.3.1 Design Considerations For Memory Cells 53
4.3.2 Design for Read operation 54
4.3.3 Design for Write operation 55
4.3.4 Transistor Sizing of memory cell 56
4.4 Sense Amplifier Design 56
4.4.1 Case Studies 57
4.4.1.1 Alpha Latch 57
4.4.1.2 Latch-type voltage sense amplifier 58
4.4.1.3 Clamped Bitline Sense Amplifier 59
4.4.1.4 Results 59
4.4.2 Design of Alpha Latch Voltage Sense Amplifier 61
4.4.2.1 Transistor Sizing of Alpha Latch 62
4.4.3 Conclusion 64
4.5 Design for generation of Sense Enable signal 64
4.5.1 Case Studies 64
4.5.1.1 Inverter Chain Scheme 65
4.5.1.2 Replica Bitline scheme based on capacitance ratioing 66
4.5.1.3 Results 67
4.5.2 Design of Replica Bitline Scheme based on Capacitance Ratioing 68
4.5.2.1 Design of dummy memory cells 69
4.5.3 Conclusion 70
4.6 Column Multiplexer 70
4.6.1 Design of the column multiplexer 71

4.7 Write Drivers 72
4.8 Conclusion 73

5 Testing and Verification **74**
5.1 Memory Cell Testing 74
5.1.1 RC interconnect delay of bitlines 76
5.1.2 Writing to a Memory Cell 77
5.1.3 Reading from a Memory Cell 78
5.1.4 Effects of varying bitline voltage 80
5.1.5 Conclusion 82
5.2 Row Decoder Testing 82
5.2.1 RC interconnect delay of word lines 83
5.2.2 Results 84
5.3 Sense Amplifier Testing 84
5.3.1 Test Setup 84
5.3.2 Results 85
5.3.3 Conclusion 86
5.4 Testing of the 4K design 87
5.4.1 Test setup 87
5.4.2 Writing data into the memory array 87
5.4.3 Reading data from the memory array 88
5.4.4 Results 90
5.5 Conclusion 90

6 Conclusion **92**

A Schematics **94**

B Layout **103**

Bibliography **111**

List of Tables

4.1 Comparison of Decoding schemes with and without the replica bitline reset signal 48
4.2 Comparison of Sense Amplifiers based on delay through memory cells 60

5.1 Results for simulations performed for Write operation 78
5.2 Bitline Differential Voltages obtained during Read operations 80
5.3 Simulations showing effects of varying Bitline Voltages 81
5.4 Simulation Results for Row Decoder Testing 84
5.5 Effects of varying bitline voltage on delay of sense amplifier 86
5.6 Effects of process variations on delay of sense amplifier 86
5.7 Final results obtained from simulations 91

List of Figures

1.1 Basic RAM architecture 2

2.1 Block Diagram of basic RAM 5
2.2 Divided Word Line Architecture (dwl) 6
2.3 Hierarchical word decoding architecture (hwd)[34] 7
2.4 Storage Density[34] 7
2.5 Memory Cell Types 8
2.6 Block Diagram of 8 to 256 decoder 10
2.7 NAND Gates 11
2.8 Sense Amplifiers 12

3.1 Critical Path of 3-tier Decode Structure 15
3.2 SRCMOS resetting techniques 17
3.3 DRCMOS resetting technique 17
3.4 Comparison of Delay for different Skewing Techniques 18
3.5 2-input Nand Gates for Pulsed Decoders 19
3.6 2 input Source Coupled Nand Gate 20
3.7 NOR style decoder [12] 21
3.8 Memory Cell Types 22
3.9 SNM for 4T and 6T cells vs cell ratio[9] 23
3.10 Methods of Sensing[46] 24
3.11 Typical Circuit for Static Voltage Sense Amplifier 26
3.12 Typical Circuits of Dynamic Voltage Sensing[46] 27
3.13 Current Sensing Principle[46] 28
3.14 Type A Current Sensing[46] 29
3.15 Type B Current Sensing 30
3.16 Type C Current Sensing 30
3.17 Seevinck's circuit[wicht35] 31
3.18 Type D Current Sensing 32
3.19 Common sense enable generation schemes 32

3.20 Variations in generation of sense amplifier enable using inverter chain scheme . . . 33
3.21 Replica Column with bitline capacitance ratioing[2] . . . 34
3.22 Working of Bitline Capacitance Ratioing Scheme[2] . . . 35
3.23 Current Ratio Based replica scheme[2] . . . 37
3.24 Skewed Word Line Driver[2] . . . 38
3.25 Multiplexer and Sense Amplifier operating as current sink (a) and current source (b).[8] . . . 39
3.26 Idea of Multiplexer compensation.[8] . . . 39
3.27 Proposed Circuit using a double multiplexer switch.[8] . . . 40
3.28 Scheme for Low Power Writes [1] . . . 42

4.1 Block Diagram of 4K design . . . 44
4.2 Block Diagram of Decoder Structure . . . 46
4.3 Decoder Design . . . 49
4.4 NOR gate used in reset block . . . 51
4.5 Local Word Line Driver . . . 52
4.6 6 Transistor Memory Cell . . . 53
4.7 3 Transistor Design Model . . . 54
4.8 Reading data from a cell . . . 55
4.9 Writing data to a cell . . . 55
4.10 Schematic of an Alpha Latch Voltage Sense Amplifier . . . 57
4.11 Latch-type voltage sense amplifier [30] . . . 58
4.12 Clamped Bitline Sense Amplifier . . . 59
4.13 Schematic of Alpha Latch Voltage Sense Amplifier . . . 61
4.14 Inverter Chain used for sense enable signal . . . 65
4.15 Waveforms showing timing scheme used in inverter chain scheme . . . 65
4.16 Replica Bitline Schematic . . . 66
4.17 Waveforms for Replica Bitline Scheme . . . 67
4.18 Replica Bitline Scheme vs Inverter Chain Scheme . . . 67
4.19 Working of Replica Bitline Scheme . . . 68
4.20 Dummy Memory Cell . . . 69
4.21 Schematic of column multiplexer . . . 71
4.22 Schematic of Write Driver . . . 72

5.1 Setup for Memory Testing . . . 75
5.2 Writing Data into Memory Cell . . . 77
5.3 Reading Data from Memory Cell . . . 79
5.4 Memory Cell contents when Bitline voltage is 900mV and 800mV . . . 81
5.5 Setup to test working of Row Decoders . . . 82
5.6 Test setup for sense amplifier . . . 85
5.7 Writing a 1 into the memory cell . . . 88

5.8 Reading data from the memory cell 89
5.9 Simulation results 90

A.1 Schematic of SCL NOR based pre-decoder 95
A.2 Word Line Reset Circuit: Zoom 1 96
A.3 Word Line Reset Circuit: Nor gate 97
A.4 Local Word Line Driver 98
A.5 Six Transistor Memory Cell 98
A.6 Write Driver Circuit 99
A.7 Precharge Circuit for Bitlines 100
A.8 Column Multiplexers 101
A.9 Voltage Sense Amplifier 102

B.1 Layout of SCL NOR based pre-decoder 104
B.2 Layout of NOR gate used in Word Line Reset Block 105
B.3 Layout of Local Word Line Driver 106
B.4 Layout of 6T memory cell 107
B.5 Layout of Write Driver 108
B.6 Layout of Voltage Sense Amplifier 109
B.7 Layout of Column Multiplexer, Write Driver and Sense Amplifier . . 110

Chapter 1

Introduction

Static random-access memory (SRAM) continues to be a critical component across a wide range of microelectronics applications from consumer wireless to high-end workstation and microprocessor applications. For almost all fields of applications, semiconductor memory has been a key enabling technology [38]. It is forecasted that embedded memory in SoC designs will cover up to 90% of the total chip area. A representative example is the use of cache memory in microprocessors. The operational speed could be significantly improved by the application of on-chip cache memory that temporarily stored a fraction of the data and instruction content of the main memory.

This thesis explores the design of SRAMs, focusing on optimizing delay and power. While process [27, 11] and supply scaling[6, 35] remain the biggest drivers of fast low power designs, this thesis investigates some circuit techniques which can be used in conjunction to scaling to achieve fast, low power operation.

The SRAM consists of an array of static memory cells which are connected by horizontal word lines and vertical bitlines as shown in Figure 1.1. To select a word line out of 2^h, a h-bit address has to be applied. The output data is usually organized as a word of b-bits. From the architectural point of view the output word represents a b-bit input/output (I/O -port). The I/O-port consists of b I/O-blocks, i.e. one block per bit of the output word. Each bit of the I/O-port can be connected to one out of 2^w bitlines by a 2^w-to-1 column or bitline multiplexer. Any SRAM cell can be accessed

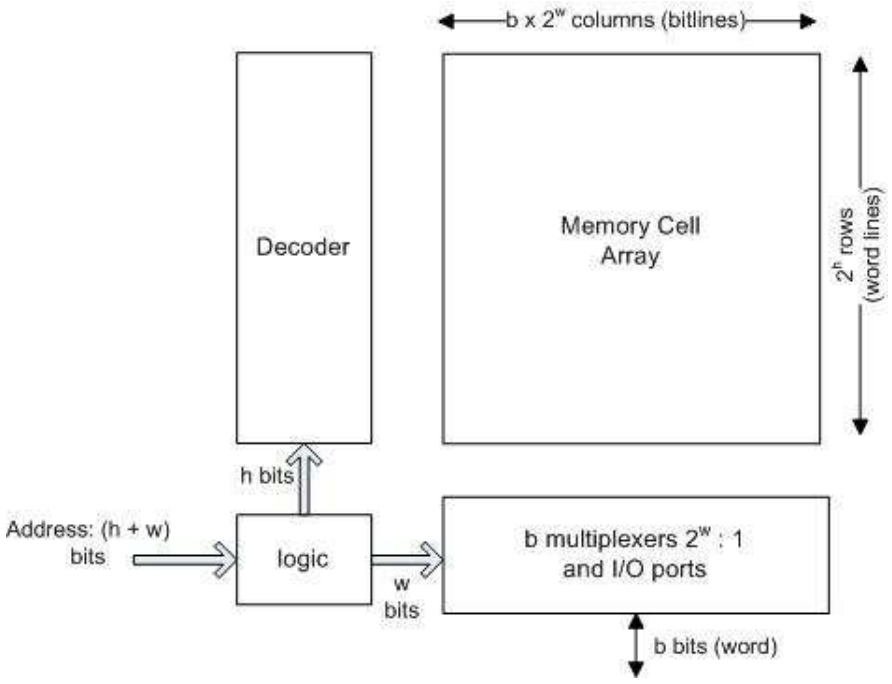


Figure 1.1: Basic RAM architecture

by an address word which is $(h + b)$ bits long. This address is applied to the control logic block which controls all the memory operations, e.g. write,read,enable,data-in,data-out.

The most common SRAM memory cell is the 6-transistor cell. It consists of two cross-coupled CMOS inverters and two n-channel transfer transistors (also called access transistors).In a read operation, the bitlines start precharged to some reference voltage usually close to the positive supply. When word line turns high, the access nfet connected to the cell node storing a 0 starts discharging the bitline, while the complementary bitline remains in its precharged state, thus resulting in a differential voltage being developed across the bitline pair. SRAM cells are optimized to minimize the cell area, and hence their cell currents are very small, resulting in a slow bitline discharge rate. To speed up the RAM access, sense amplifiers are used to amplify the small bitline signal and eventually drive it to the external world.During a write operation, the write data is transferred to the desired columns by driving the data

onto the bitline pairs by grounding either the bit line or its complement. If the cell data is different from the write data, then the 1 node is discharged when the access nfet connects it to the discharged bitline, thus causing the cell to be written with the bitline value.

The next chapter introduces the various techniques which are used in practical SRAMs and motivates the issues addressed by this thesis. The circuit techniques used to design the three major blocks of an SRAM viz. the memory cell, the decoder design and the sense amplifiers, are further elaborated in chapter 3. Some circuit techniques to reduce the power dissipation of write operations and column multiplexer's are also mentioned in chapter 3.

The design of a 4K SRAM has been described in Chapter 4.This design incorporates some of the techniques mentioned in chapter 3 and also describes a unique technique for resetting the word line pulse width. This design has been tested and verified in chapter 5.

Chapter 2

Overview of SRAM

In this chapter the basic working of an SRAM and its major blocks, namely the memory cells, the row decoder and the sense amplifiers have been described. The different architectures that are currently prevalent in the industry are also outlined.

2.1 Operation of a basic Random Access Memory

A typical random access memory (RAM) architecture is as shown in Figure 2.1. It consists of a matrix of memory cells arranged in an array of 2^N rows by 2^M columns. The total size of the memory array is $2^M \times 2^N$ bits.

During a read operation, one of the 2^N rows (Word lines) is selected by the row address decoders by decoding the row addresses. All the memory cells in the given word line are enabled. The column decoder selects one of the 2^M columns and the value of the selected memory cell is read out by the sense amplifier. The data into and out of the memory array is controlled by the Read-Write control circuit.

2.1.1 SRAM architectures

In its simplest form the SRAM has a monolithic architecture as shown in Figure 2.1. It has only one memory array which is accessed by the row and column decoders. This architecture has major disadvantages as the size of the array grows.

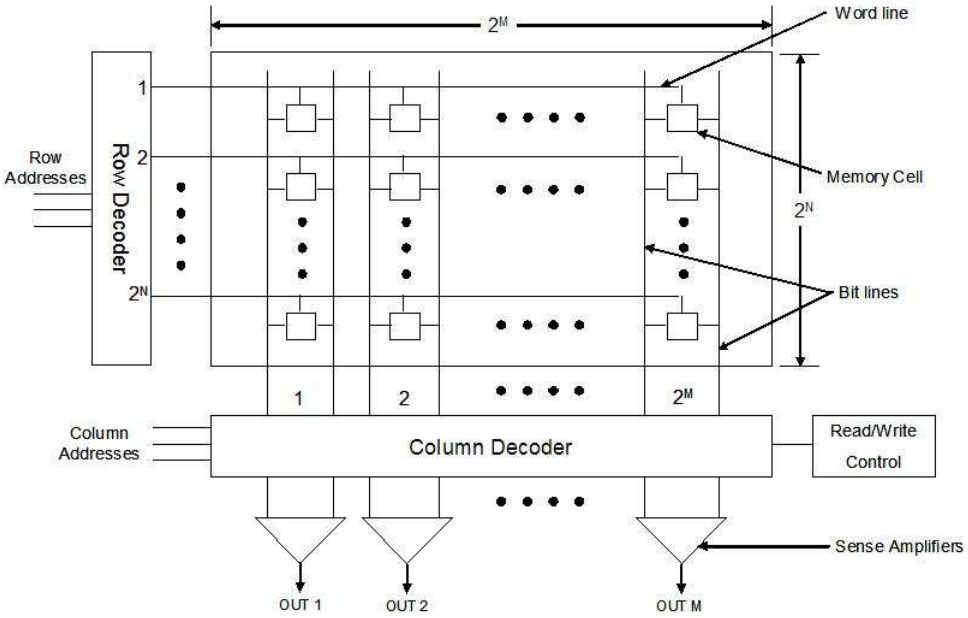


Figure 2.1: Block Diagram of basic RAM

The word line RC delay grows as the square of the number of cells in the row, and bit line power grows linearly with the number of columns. Thus for larger SRAM designs, the memory array is partitioned into sub-arrays, commonly referred to as macros. The size of each macro can vary anywhere from 16kB to 256kB. Any macro greater than 256kB is further divided into smaller sub-macros. Each macro stores a part of the accessed word, called the sub-word, and all of these sub-words are accessed simultaneously to access the complete word [29, 12, 10]. Each macro can be treated as a separate SRAM, as each has its own set of local word line drivers and sense amplifiers. They however share some parts of the decoder.

Yoshimoto et. al. [25] first proposed the Divided Word Line (DWL) technique, where the macros are further divided into smaller blocks of cells. In the DWL technique the long word line is further divided into n sections and each of these sections can be activated independently. This reduces the word line length by n and the RC delay by n^2. Figure 2.2 shows the DWL architecture where a macro of 256 columns is further divided into 4 parts of 64 columns. Each sub-part has its own set of local

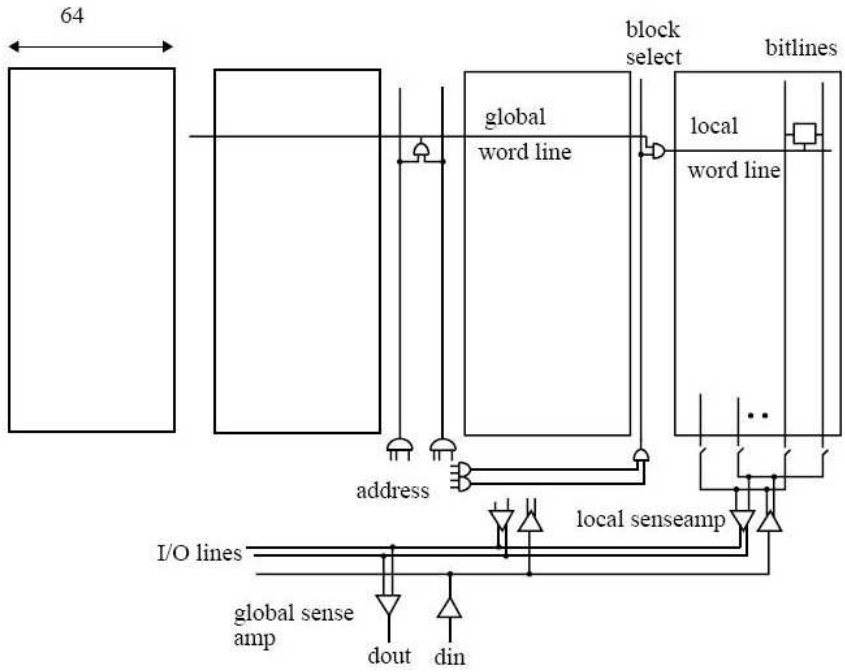


Figure 2.2: Divided Word Line Architecture (dwl)

word line drivers. A particular row is selected by first activating the global word line driver which selects a particular block, and the local word line drivers then select a particular row within this block. Thus, the local word line is now reduced to 64 columns and this reduces the RC delay by a factor of 4. Even though the Global word line is nearly as long as the width of the macro, it still has a lower RC delay than a full length word line since its capacitive loading is smaller. The global word line driver has to drive only 4 local word line drivers, instead of 256 memory cells. The word line RC delay is reduced by another factor of four by keeping the word line drivers in the center of the word line segments thus halving the length of each segment. The column current is also reduced by a factor of four, since only 64 memory cells are activated at a time instead of all 256.

This concept of DWL was further evolved by [34] to what is known as the Hierarchical Word Decoding (HWD) technique. The HWD technique is similar to the DWL but it divides the blocks into sub-blocks, and so a three tier decoding hierarchy or

more can exist[34]. For 4Mb or larger SRAM sizes the HWD architecture has become very popular.

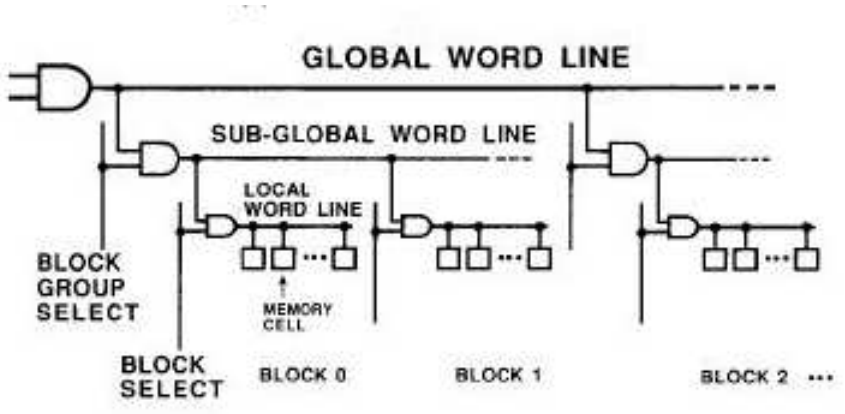


Figure 2.3: Hierarchical word decoding architecture (hwd)[34]

The number of hierarchies, in other words, the level of word-line division, is determined by the total load capacitance of the word decoding path. In Figure 2.3 there are three hierarchies, consisting of the Global Word Line, the Sub-Global Word Line and the Local Word Line. Hirose et. al.[34] also showed results Figure 2.4 which show

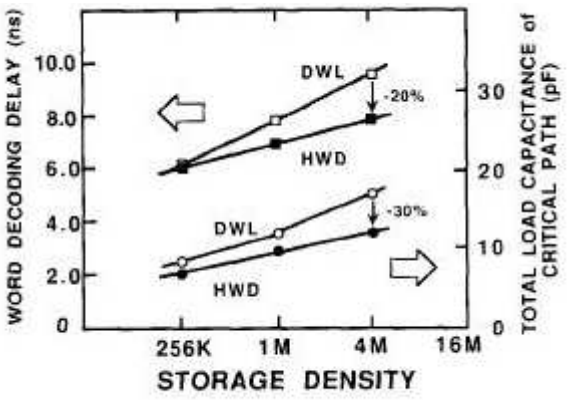


Figure 2.4: Storage Density[34]

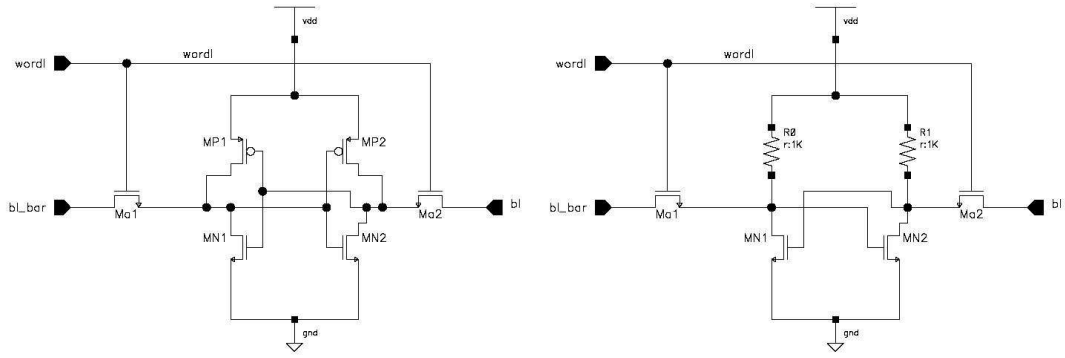
that for small memory densities in the order of 256K there is no significant difference

between the DWL and the HWD architectures. But as the memory densities start increasing there is a significant advantage in using the HWD architecture.

2.1.2 Circuit Techniques in SRAM design

From the architectures it is evident that the design of an SRAM consists of three major blocks, the design of the memory cell, the decoder circuits and the sense amplifiers. In the following sections, we will have a brief overview of each of these blocks.

2.1.2.1 Memory Cell Types



(a) Six Transistor Memory Cell (b) Four Transistor Memory Cell

Figure 2.5: Memory Cell Types

The two most commonly used memory cell architectures are the 6T and the 4T memory cells Figure 3.8. While the 4T memory cell has a smaller area it also has lower stability as compared to the 6T cell [9]. The 6T cells have a much better Static Noise Margin (SNM) over the 4T cells.

This is especially true for lower voltage levels. In order to maintain stability the 4T cells have to maintain higher cell ratios compared to the 6T cells, and this has eroded some of the cell size advantages that they had. Also, as pointed out by [9] the 6T cells have historically been more robust to Soft Error rates (SER) as compared

to the 4T cells. However, with the 6T cells also being scaled down to smaller sizes and with the reducing voltages, SER will become a significant issue in 6T cells as well. The working of a 6T cell has been described in many publications [23, 4]. MP1 ,MP2, MN1 and MN2 form the cross-coupled inverter pair that stores the data. Ma1 and Ma2 are the access transistors, which allow data to be written or read from the cell. The gates of Ma1 and Ma2 are controlled by the word line and their drains are connected to the two bitlines. For the 4T cells, the pMOS transistors MP1 and MP2 are replaced by resistors R0 and R1.

2.1.2.2 Row Decoders

Row decoders are used to assert the word lines based on the input addresses. The decoders are designed to drive the capacitance of the word lines and have small power dissipation. Logically the decoder is equivalent to 2^n n-input AND gates where the AND operation is performed in a hierarchical manner. The decoder structure mainly consists of an initial predecoder stage, where a group of address inputs are decoded to give the predecoder outputs. These are then combined at the next stage to drive either the next stage of decoders or the word lines .

Figure 2.6 shows an 8 to 256 decoder structure. In the predecoder stage, the address inputs A0 - A3 and their compliments $\overline{A0 - A3}$ are combined using a 4 to 16 predecoder. The two sets of 16 predecoded outputs, are then combined to give the 256 outputs which drive the word lines. The decoder delay mainly consists of the gate delays in the critical path and the RC delay of the predecoder and word line wires. The delay and the power of the decoder gates can be reduced by sizing the transistors. [39, 41] explore sizing of the gates to reduce delay, while [15, 5]do so for low power. The circuit style can also affect the delay caused by the decoders. [29, 10, 12] have shown that a pulsed decoder greatly reduces the delay as compared to using a conventional decoder. This is because; in a conventional decoder one of the 2^n outputs will be asserted based in the input address. For the next set of address inputs, this output has to be first de-asserted, and the new decoded output has to be asserted. Thus, the decoder gate delay in such a design is the maximum of the

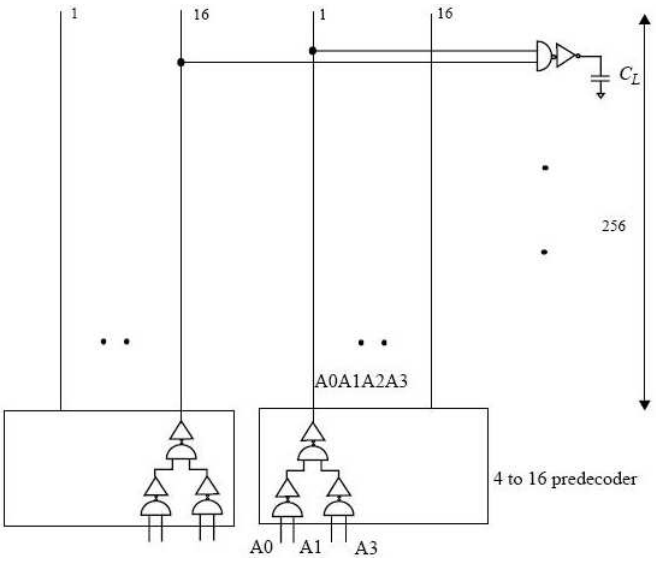
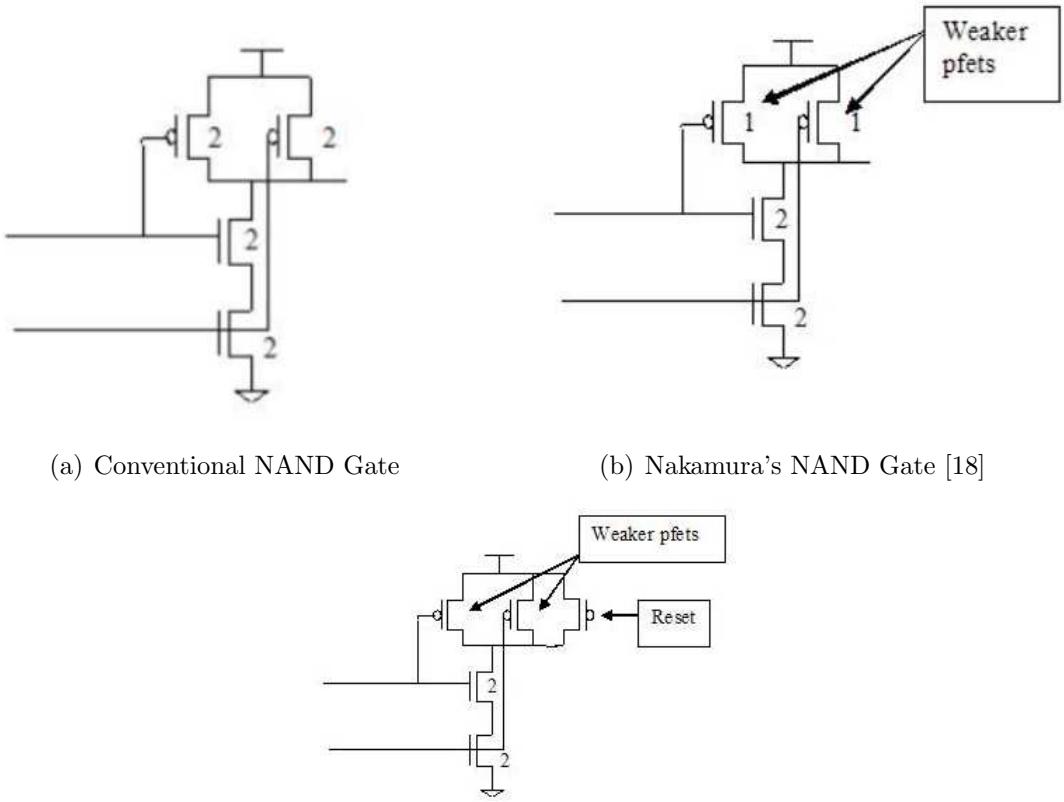


Figure 2.6: Block Diagram of 8 to 256 decoder

delay to de-assert the old word line and the delay to assert a new word line, and it is minimized when each gate in the decode path is designed to have equal rising and falling delays. In a pulsed decoder, the decoder output stays active for a certain minimum time and then shuts off. Thus, before any access all the word lines are off, and the decoder needs to just activate one of the word lines. Since there is only one kind of transition taking place, the decoders can be skewed to achieve greater speed as shown in figure 2.7. As can be seen in figure 2.7(b), the pMOS size if halved as compared to the traditional NAND gate. This does not affect the rising delay, since both the inputs are guaranteed to be de-asserted. This reduces the loading of the previous stage and reduces the overall decoder delay. Figure 2.7(c) [29] shows another implementation where the de-assertion of the gate is completely independent of its assertion. SRCMOS and DRCMOS techniques have been used to reset the decoders externally.

The n-input AND function can be implemented via different combination of NANDs, NORs and Inverters. Since in current CMOS technologies, a pfet is at least two times slower than an nfet, NORs are inefficient and so the AND function is best achieved by a combination of NANDs and inverters. In chapter 3 we shall look



(a) Conventional NAND Gate

(b) Nakamura's NAND Gate [18]

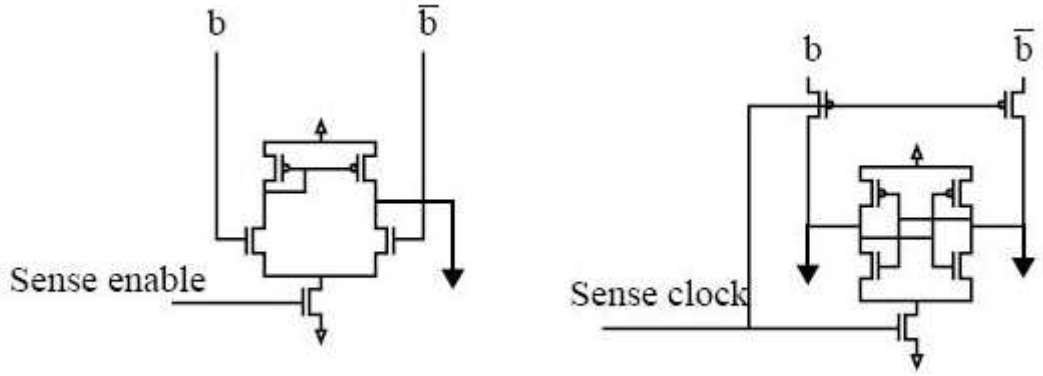
(c) Skewed NAND Gate

Figure 2.7: NAND Gates

at the basic styles used to build a NAND gate such as the conventional series stack NAND, the Source coupled NAND and the NOR style NAND.

2.1.2.3 Sense Amplifiers

The sense amplifiers have to amplify the data which is present on the bitlines during the read operation. The memory cells are weak due to their small size, and hence cannot the discharge the bitlines fast enough. Also, the bitlines continue to slew till a large differential voltage is formed between them. This causes significant power dissipation since the bitlines have large capacitances. Hence, by limiting the word line pulse width we can control the amount of charge pulled down by the bitlines and



(a) Typical Linear Type Sense Amplifier (b) Typical Latch Type Sense Amplifier

Figure 2.8: Sense Amplifiers

hence limit power dissipation [26, 17, 28]. Also, as shown by [2] we can use external structures, such as replica bitlines, which can control the word line pulse width to be just wide enough for a differential voltage to be developed which can be read reliably by the sense amplifiers. The sense amplifiers are designed to amplify a differential voltages of 100mV.

There are two main categories into which sense amplifiers can be classified: the linear amplifier type [20] and the latch amplifier type [29, 12, 10]. Figure 2.8(a) shows a simple prototype of the linear type amplifier. This needs a DC bias current, to set it up in the high gain region, before the sensing operation is performed. It needs two or more stages to effectively convert the small bitline swing into a CMOS level signal. These types of amplifiers are preferred for low power and low voltage design, since they consume biasing power and operate over a limited supply voltage.

Figure 2.8(b) shows the basic prototype of a latch type sense amplifier. It consists of two cross coupled gain stages which are enabled by the sense clock signal. The cross coupled stage ensures a full amplification of the input signal. This type of amplifier consumes least amount of power, however they can potentially be slower since some timing margin is needed for the generation of the sense clock signal. If the sense amplifier is enabled before sufficient differential voltage is formed, it could lead to a

wrong output. Thus, the timing of the sense clock signal needs to be such that the sense amplifier can operate over various process corners and temperature ranges.

2.2 Conclusion

In this chapter we have outlined the architectures which are prevalent in SRAM design. Some of the key components of an SRAM design include the decoders, the memory cells and the sense amplifiers. A brief overview of the circuits techniques which are used in designing these components has been outlined. In the next chapter we have a detailed description of these techniques which have been cited in literature.

Chapter 3

Fast Low Power SRAM Decoder and Data Path Design

In this chapter, we will explore the design of fast low power decoders and the data path. We will look at the design considerations for the decoders, the memory cells and the sense amplifiers. At the end of the chapter we will look at certain techniques which reduce the power dissipation during write operations and some compensation techniques for column multiplexer's.

3.1 Design of Fast Low Power Decoders

We will look at how to reduce the logic effort of the decoders using skewing and circuit techniques.As shown in [29, 10, 12] pulsed signals are preferred over level signals as they have less power dissipation. As shown in [29] using pulses to activate the word line reduces the power consumption while also reducing the bitline access path. Figure 3.1 shows the critical path of the decoder in a typical SRAM based on the DWL architecture. It consists of the address buffers followed by the predecoders which in turn drive the global word drivers, followed by the local word drivers. All these stages have their inherent RC-delays.

The decoding problem can thus be viewed as a combination of two tasks: Determining the optimum decode hierarchy and choosing the best circuit styles to obtain

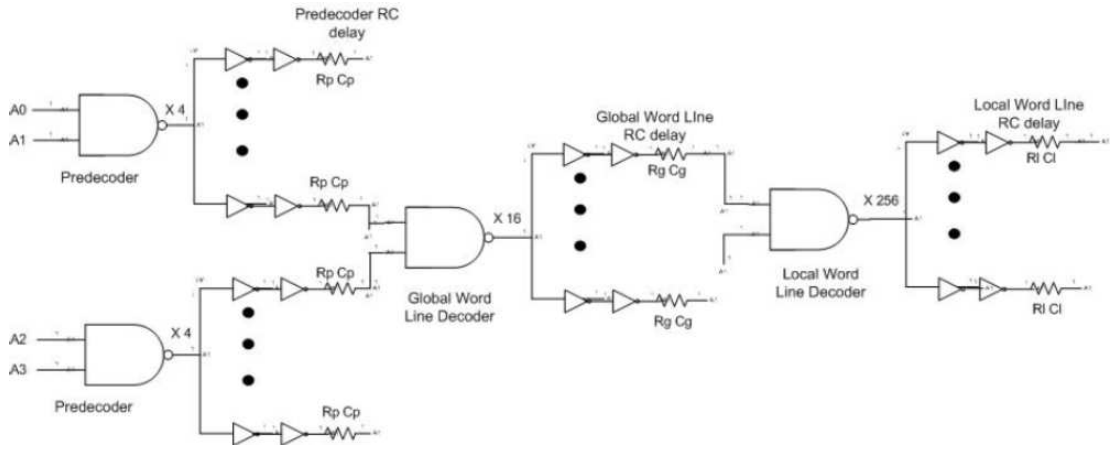


Figure 3.1: Critical Path of 3-tier Decode Structure

low power and least delay. The problem of optimally sizing a chain of gates for optimal delay and power is well understood in [39, 41, 15, 5].

3.1.1 Sizing and Buffering Considerations

As seen in Figure 3.1, the critical path of the decode structure has multiple chains of gates separated by wires having RC delays. An important aspect in the design of the decoder is to determine the optimum size of it. When the delay needs to be minimized, we can derive the optimum sizing criterion using some simple models for the gate delays. We are assuming that the interconnect is fixed and independent of the decoder design. In [3] the various sizing techniques and their impact on the delay, area and energy of the decoder have been studied. The sizing techniques for minimum delay and minimum power have also been derived in [3]. First the sizing of an inverter chain is studied for minimum power and delay, and the concepts are then applied to the design of decoders. The three key features of the decode path which distinguishes it from that of a simple inverter chain is the presence of logic gates other than inverters, branching of the signal path at some of the intermediate nodes, and the presence of interconnect inside the path. Logic gates and branching are easily handled by using the concept of logical effort and branching effort introduced in [44].

It has been proved that a fan-out of 4 can be used as a good heuristic when sizing the decoders.

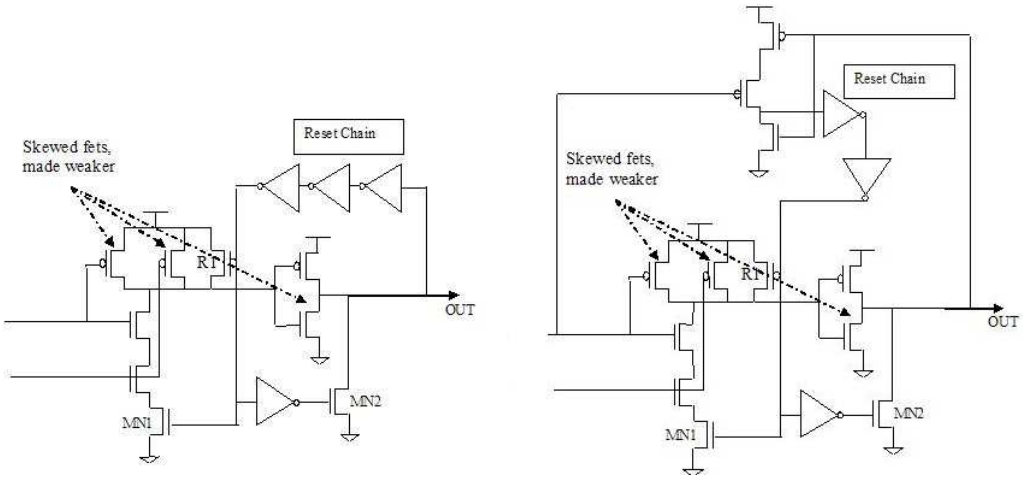
3.1.2 Circuit techniques involved in decoder design

The logical effort of the entire decode path can be reduced in two ways 1) One by skewing the fets and 2) Using different circuits to generate an n-input AND gate. We first describe the benefits of skewing the gates and show techniques of implementing them in a power efficient way. Then various circuit topologies are discussed which help in reducing the delay of the decode path. Finally, the various tradeoffs are discussed in the circuit techniques and a comparative study is done based on the power consumed and the delay involved in driving a word line containing a specified set of memory cells.

3.1.2.1 Reducing the Logical Effort

Since the word lines are asserted only in a particular direction, by skewing the gates in that direction, we can reduce the delay as well as the capacitance of the inputs. The various techniques that are present involve resetting the decoder using either some internal or external gates. The pre-charge logic decoders use the clock to do this, but this method consumes more power since the clock has to pre-charge all the gates and then reset all of them. The SRCMOS [29, 13] uses self-resetting techniques to achieve this. However, care has to be taken to make sure that the decoders do not reset too early, else the word line pulse width is too short to enable the memory cell. The DRCMOS [37, 12] decoders use a delayed version of the output to reset themselves.

Figure 3.2(a) shows a NAND gate using SRCMOS resetting technique. One of the advantages of using the SRCMOS decoder is that only the gate which has been activated is reset, unlike in the pre-charge logic gates where all the gates are reset by the clock input.When the decoder is not asserted, the OUT signal is low. At this point the reset pMOS transistor R1 is disabled. When the decoder is enabled, the OUT signal goes high. This signal then travels through the reset chain and switches



(a) self-reset (b) Predicated Reset Technique

Figure 3.2: SRCMOS resetting techniques

ON the reset transistor R1 and switches OFF MN1. Thus, the decoder is reset and the pulse width is equal to the delay through the reset chain. The reset chain needs to be designed carefully to achieve a minimum word line pulse width. Another technique shown in [13] uses a predicated reset chain Figure 3.2(b). Using this predicated reset chain approach ensures that the delay through the reset chain is longer than the input pulse width.

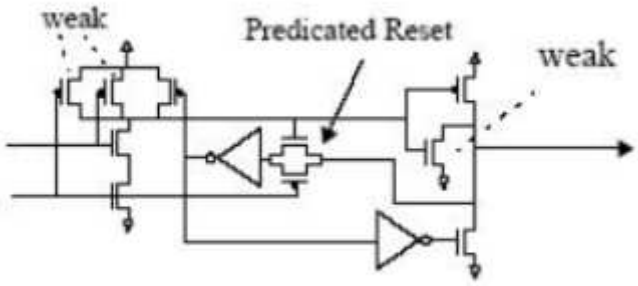


Figure 3.3: DRCMOS resetting technique

Figure 3.3 shows the DRCMOS resetting technique. The DRCMOS has the advantage that it has a predicated reset and does not need the extra nfet needed in the SRCMOS technique. The DRCMOS technique thus has the lowest logic effort and least delay. The only problem is that the output pulse width is larger than the input pulse width.

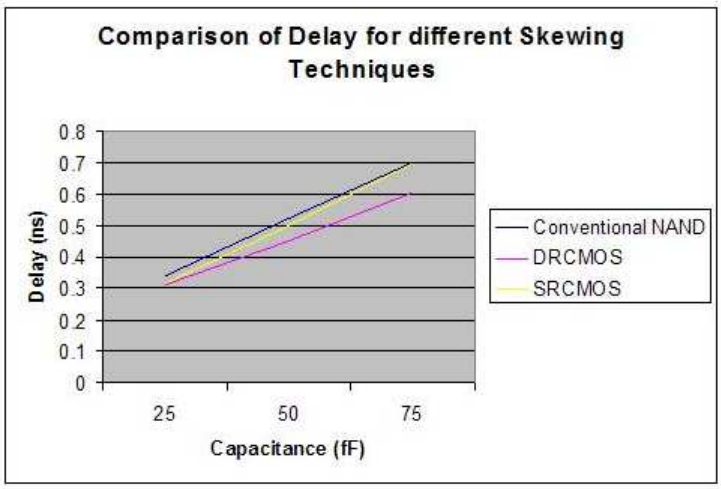


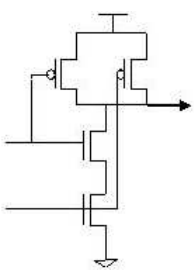
Figure 3.4: Comparison of Delay for different Skewing Techniques

Three 4-to-16 decoders were simulated using the conventional NAND, SRCMOS and DRCMOS skewing techniques. Capacitive loads ranging from 25fF to 75fF were used to compare the delay through each scheme. The graph 3.4 plots the results that were obtained from the simulations. As can be seen the DRCMOS technique has the least delay over a wide range of capacitances. A major problem with the DRCMOS technique is that the output pulse is larger than the input pulse, and hence it cannot be used in successive stages. Thus, using DRCMOS gates broken by SRCMOS gates is the best strategy for designing a decoder. This prevents the output pulses from becoming too wide. A main constraint to this scheme would be the area overhead, as these schemes involve using a lot more fets. Also, in order to control the pulse width it is best to use a SRCMOS based gate in a preceding decoder block such as the global driver or block select driver and to use the conventional driver in the final

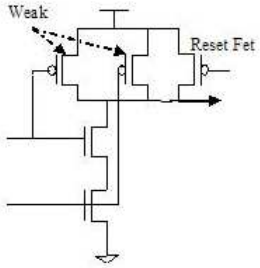
stage. Another scheme to control the pulse width has been used in the design of the 4k macro described in chapter 4, where the output from a replica bitline is used to reset the local driver and thus control the pulse width. As has been shown by [3] the maximum speedup which can be achieved by using the skewing technique is $\sim 40\%$ as compared to an unskewed scheme. We will next look at the different circuits which can be used to obtain an n-input AND gate operation.

3.1.2.2 Different decoder design schemes

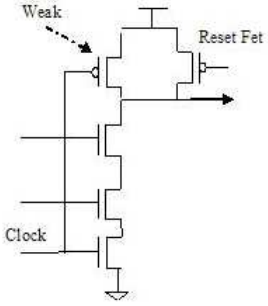
In this section we will look at the different schemes of making an n-input AND gate. In the last section we saw how skewing the gates can reduce delay by up to 40% and in this section we will see how to reduce the power consumption as well as the delay by using different circuits.



(a) Conventional Nand



(b) Skewed Nand



(c) Clocked Nand

Figure 3.5: 2-input Nand Gates for Pulsed Decoders

The simplest circuit type to build a NAND gate, is the series stack conventional

NAND gate shown in Figure 3.5(a). In pulsed decoders since both the inputs are guaranteed to fall low, we can skew the fets and still maintain the same delay [18] as before. This reduces the logical effort for the conventional static gate [3]. It is best to use 2-input gates as this reduces the nmos vertical stack which leads to a reduction in the capacitance and logical effort. Thus, the gates with 3-inputs or more are best made by using cascades of the 2-input gates.

We can also use the domino style gating since the clock is needed for the input stages of the pulsed decoders. This is shown in Figure 3.5(c). The pmos which is driven by the clock can be made weak, while the nmos fets in the stack are made progressively larger. The advantages of such circuits are that they are easy to build, are robust and also have a good noise margin. However, we can achieve similar NAND gate functionality by using fewer fets using the SCL circuits as shown next.

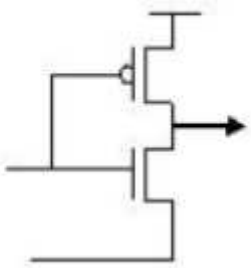


Figure 3.6: 2 input Source Coupled Nand Gate

Figure 3.6 shows a 2-input Nand gate implemented in the source coupled style [20, 24]. As compared to a conventional series stack Nand gate two fets are eliminated. This gate can be as fast as an inverter if the branching effort of the source input is large and if the total capacitance on the source line is much greater than the output load capacitance for the gate [3].

One of the candidates for building high speed predecoders is the NOR based NAND gate. One of the major advantages of this circuit is that nfets are added in parallel as the number of inputs go up. This keeps the logical effort constant, unlike

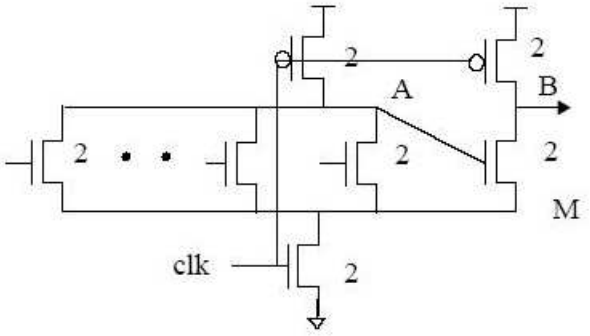


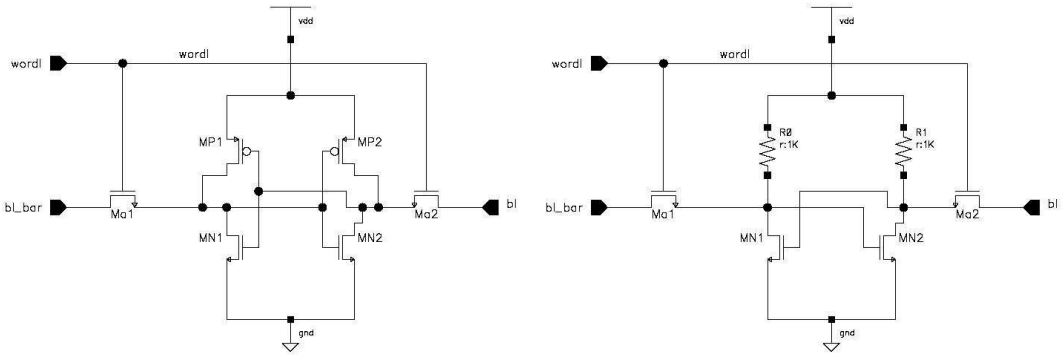
Figure 3.7: NOR style decoder [12]

the previous two circuit styles. Figure 3.7 shows an implementation of the NOR based NAND gate by Nambu et.al [12]. For any number of inputs the logical effort remains constant, since only three fets are ever connected in a vertical stack.The nfet (M) shares the same source as the input fets, however it's gate is controlled by the output of the NOR gate. When clock (clk) is low, both nodes A and B are precharged high. When clock goes high, the behavior of the gate depends on the input values. If all the inputs are low then node A remains high while node B discharges and the decoder output is selected. If any of the inputs are high, then node A discharges, shutting off M and preventing node B from discharging and hence causing that output to remain unselected. As this situation involves a race between A and B, the gate needs to be carefully designed to ensure robust operation.

3.2 Memory Cell Types

The two most commonly used memory cell architectures are the 6T and the 4T memory cells (Figure 3.8). While the 4T memory cell has a smaller area it also has lower stability as compared to the 6T cell [9]. As shown in Figure 3.9 [9] the 6T cells have a much better Static Noise Margin (SNM) over the 4T cells.

This is especially true for lower voltage levels. In order to maintain stability the 4T cells have to maintain higher cell ratios compared to the 6T cells, and this has



(a) Six Transistor Memory Cell (b) Four Transistor Memory Cell

Figure 3.8: Memory Cell Types

eroded some of the cell size advantages that they had. Also as pointed out by [9] the 6T cells have historically been more robust to Soft Error rates (SER) as compared to the 4T cells. However, with the 6T cells also being scaled down to smaller sizes and with the reducing voltages SER will become a significant issue in 6T cells as well. The working of a 6T cell has been described in many publications [23, 4]. MP1,MP2, MN1 and MN2 form the cross-coupled inverter pair that stores the data. Ma1 and Ma2 are the access transistors, which allow data to be written or read from the cell. The gates of Ma1 and Ma2 are controlled by the word line and their drains are connected to the two bitlines. For the 4T cells the pMOS transistors MP1 and MP2 are replaced by resistors R0 and R1.

3.2.1 Mechanism of Cell Instability

The schematic of the 6T transistor is as shown in Figure 3.8(a). One of the reasons for the cell instability is if the related pMOS starts conducting, when a zero is stored in the cell.

Assume that a logical one is the stored value i.e. bit equal to 1 and $\overline{bit}$ equal 0. The 0 stored in $\overline{bit}$ will not be weakened as long as

$$V_{cell} \geq V_{DD} - V_{thp} \tag{3.1}$$

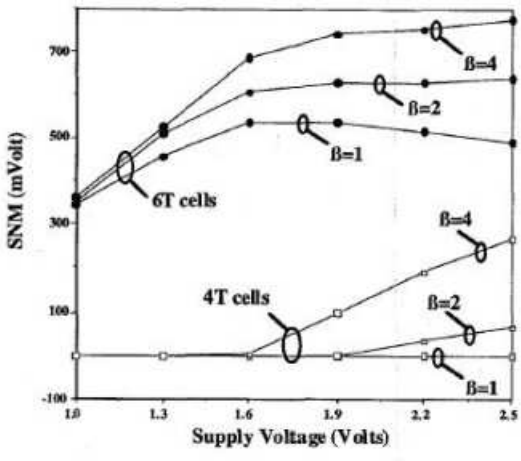


Figure 3.9: SNM for 4T and 6T cells vs cell ratio[9]

where, V_{cell} is the internal cell voltage. The voltage V_{cell} basically decreases due to the voltage divider formed by Ma1 and MP1. The opposite case is not that critical since it is determined by the series connection MN2/ Ma2 between V_{BL} and ground. In a correctly designed memory cell the nMOS transistors MN1, MN2 are strong enough that the node voltage associated with a logical zero never exceeds V_{thn}.

A violation of the stability criterion given by (3.1) can cause MP2 to turn on. This increases the voltage V_{cellB} associated with the logical 0 ($\overline{bit}$).This reduces the current i_{cell} through the series connected transistors MP2 and Ma2 and it slows down the reading. It has been has shown that as long as the cell does not reach metastability ($V_{cell} = V_{cellB}$) the contents of the cell are not lost. [46] have run simulations for the 6T cell in 180nm technology with $V_{DD} = 1.8$V and have shown that if the bitline voltage (V_{BL} falls below 0.68V, the cell contents are lost. Both the storage nodes fall to a common value and the cell restores a random value after the access transistors are deselected. A critical bitline voltage has been derived by [46] as shown

$$V_{BL,crit} = (V_{DD} - V_{tht}) - \sqrt{\frac{\beta_p}{\beta_n}(2V_{DD} - 3V_{thp})V_{thp}} \tag{3.2}$$

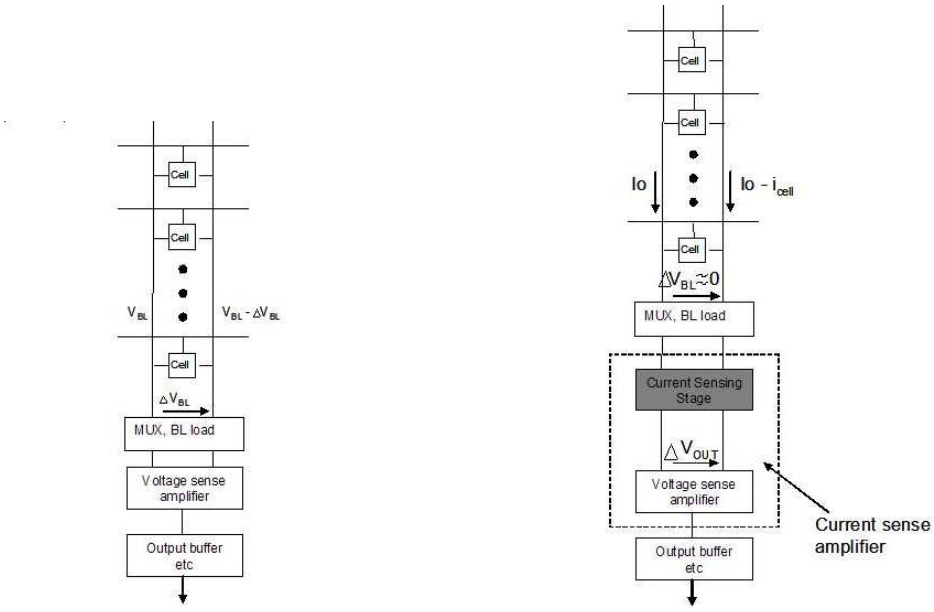
In order to maintain performance and stability of the memory cell it has to be ensured that V_{BL} never falls below $V_{BL,crit}$.

3.3 Sense Amplifiers

It has been shown in [46] in order to achieve high speed operation in low voltage applications, current sensing is very promising. In this section, we shall show both the voltage and current sensing techniques that are used in academia and industry. In chapter 4 we have shown using case studies, how current sensing is faster than voltage sensing. This is due to the fact that current sensing avoids large voltage swings of the highly capacitive bitlines. This however comes at the cost of added power dissipation.

3.3.1 Methods of sensing

During a read operation the contents of a memory cell are read and amplified by using sense amplifiers. Both voltage and current sense amplifiers can be used. Figure 3.10(a) shows the typical setup for a voltage sense amplifier while Figure 3.10(b) shows the typical setup for a current sense amplifier. Bitline loads and



(a) Setup for Typical Voltage Sensing

(b) Setup for Typical Current Sensing

Figure 3.10: Methods of Sensing[46]

precharge circuits are used to make sure that before sensing, both the bitlines are pulled high. The world line for the memory cell to be read is then activated and the bitline load is deactivated. A small current is drawn into the selected cell on that side of the cell where the 0 is stored. This causes a voltage swing ΔV_{BL} at the bitline and thus the bitline potential reduces to $V_{BL} - \Delta V_{BL}$. The other side of the memory cell contains a 1 and thus there is no voltage drop on that bitline. This change in voltage is read by the voltage sense amplifier and it generates a CMOS level output signal. The major disadvantage of this principle is that it is relatively slow because of the large bitline capacitance which has to be discharged.

However, conventional SRAM designs still use voltage sensing because it is a well-known structure, has a clear design effort and also has acceptable area and power consumption. In a current sensing sense amplifier, a current-mode amplifier is inserted in front of the voltage sense amplifier Figure 3.10(b). The current stage now takes the current $I_o - I_{cell}$ and gives a voltage as an output, which is evaluated by the voltage sense stage. Thus, the current sensing circuit acts as a buffer stage that decouples the voltage sense amplifier from the highly capacitive bitline. Due to this buffer stage, only a small signal voltage swing occurs at the bitline. This causes the ΔV_{out} to develop a lot faster than ΔV_{BL} in conventional voltage sensing since the output of the current sensing stage does not see a large load capacitance.

3.3.2 Voltage Sense Amplifiers

In this section we will first look at the basic principles and typical circuits which are used as voltage sense amplifiers.

3.3.2.1 Static Voltage Sense Amplifiers

The most commonly used static voltage sense amplifier is shown in Figure 3.11[12, 43]. The amplifier is only enabled during the read operation by using the Sense Enable line. This signal can be generated using different schemes which are discussed in a later section. Due to noise and mismatch the sense amplifier might start to develop a wrong output signal. A major advantage of static sense amplifiers is that they have

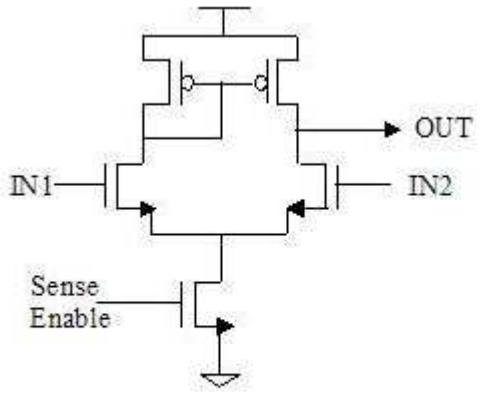


Figure 3.11: Typical Circuit for Static Voltage Sense Amplifier

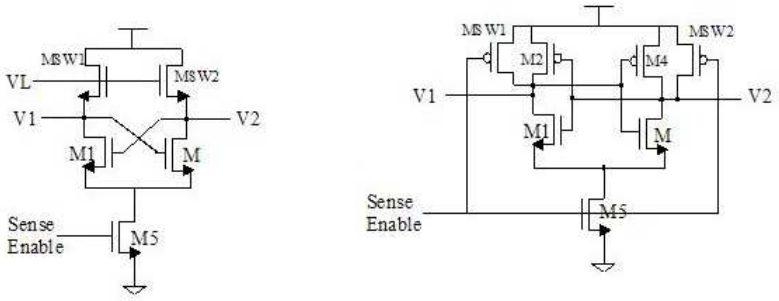
high immunity to noise and are able to recover and give the correct output if the input becomes large enough. A major disadvantage of this scheme is its speed which is determined by the time constant [46]

$$\tau_{static} = r_{DS} C_L \tag{3.3}$$

Where r_{DS} is the drain-source resistance of the transistors at the output node and C_L is the parasitic capacitance. Multi-stage amplifiers can be implemented to increase speed [12, 30]. Weak positive feedback can also be used [36] to increase speed.

3.3.2.2 Dynamic Voltage Sense Amplifier

These circuits are also referred to as latch-type sense amplifiers since they use strong positive feedback causing latching behavior. Figure 3.12 shows the typical circuits of dynamic voltage sense amplifiers. As shown in Figure 3.12(a) Sense Enable is used to enable the amplifier. The amplifier is enabled when a sufficient voltage difference is available between the V1 and V2 nodes. These nodes act as both the input and the outputs. The classical sensing latch has become popular since it can be used in DRAMs to restore data in addition to sensing. The Figure 3.12(b) is the most commonly used voltage sense amplifier [14, 32]. It is based on cross-coupled CMOS inverters where the output of one inverter controls the input of the other forming a



(a) Classical Sensing Latch (b) Conventional CMOS latch circuit

Figure 3.12: Typical Circuits of Dynamic Voltage Sensing[46]

fully symmetrical structure.

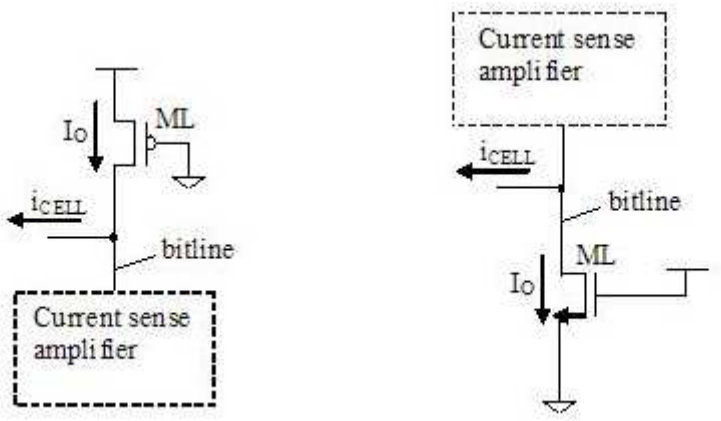
3.3.3 Current sense amplifiers

In this section, we will look at the advantages that Current sense amplifiers offer over voltage sense amplifiers. We will look first look at the classification of current sense amplifiers [46, 43]. Based on [46] a comparison is done between these different classes of current amplifiers, based on area, power etc.

3.3.3.1 Basic working principle of current sense amplifiers

Figure 3.13 shows the working principle of a current sense amplifier. A bitline load ML is used, which along with write recovery also provides a bias current. This static bias current improves performance by making the sensing faster, however this also increases the power dissipation. A current sense amplifier can be used in both sink as well as source mode. All current sense amplifier designs are based on feedback structures with the sensing transistor placed in the feedback loop. This is done so as to have low input impedance.

The most common current sense amplifier architecture is the sink type, since it is similar to the conventional voltage sensing setup, from where it emerged. One of the major design principles is to make sure that the bias current is not too small since



(a) Current Sensing in Sink Mode

(b) Current Sensing in Source Mode

Figure 3.13: Current Sensing Principle[46]

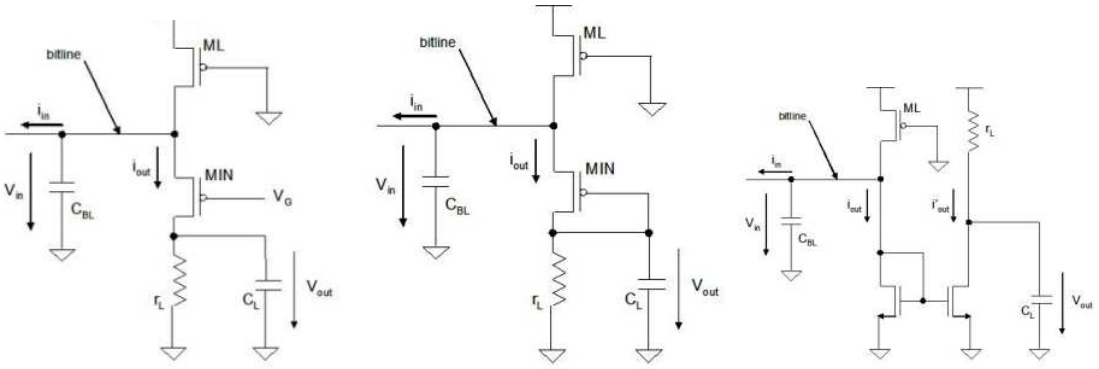
the cell current is subtracted. So in order to ensure correct operation, we need to make sure that

$$I_o > I_{cell} \tag{3.4}$$

where I_{cell} refers to the cell current. Sensing will fail if (3.4) is violated since the circuit will then have a high-impedance input. The bitline is discharged by the current difference $I_{cell} - I_o$. We will next look at the four different classes of current sense amplifiers in brief.

3.3.3.2 Type A - Cascode and Diode Connection

Figure 3.14 shows the Type A class of current sense amplifiers [36]. Cascode circuits are often used due to their low input resistance, which is defined by the transconductance of the common-gate transistor. The drain of this transistor represents the output current i_{out}. This output current can be converted to a voltage v_{out} by using a load resistance r_L. The simplest case of this type of amplifier is shown in Figure 3.14(b) where the gate voltage V_G is set to ground potential. However, it has to be ensured that the transistor stays in saturation [20]. Another implementation is shown in Figure 3.14(c) where a current mirror is used to generate the output voltage



(a) cascode (b) cascode with Gate grounded (c) diode-connected input transistors

Figure 3.14: Type A Current Sensing[46]

v_{out}[24]. One of the disadvantages of using such a circuit is that is had a single ended input, and thus two such circuits are needed for the complementary bitlines. Type B to D are more suitable for differential current sensing as described in the following sections.

3.3.3.3 Type B - Regulated Cascodes

As shown in Figure 3.15 the Type B current sense amplifier is a modification of Type A. It works on the principle known as regulated cascade where the gate potential is controlled by a negative feedback from the input node. [45] have used the above principle for general current detection. The feedback forces the bitline to become V_{ref}. When the voltage gain is greater than zero, the amplifier provides a negative feedback from the input node to the gate of the input pMOS. Differential sensing can also be performed on a similar principle and thus both the complementary bitlines can be sensed using this scheme.

3.3.3.4 Type C - Negative Impedance Converter

Figure 3.16 shows the Type C current sense amplifier. Type C amplifiers use negative feedback instead of positive feedback. Here instead of the input voltage, the

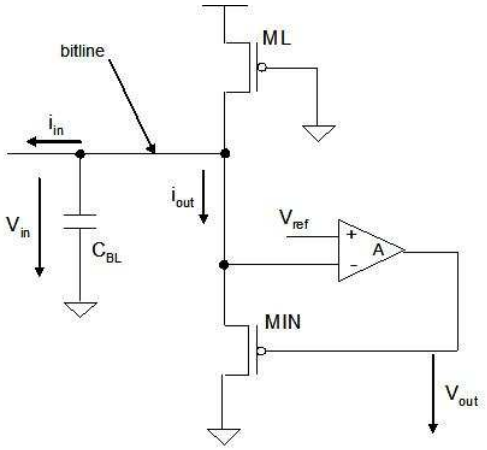


Figure 3.15: Type B Current Sensing

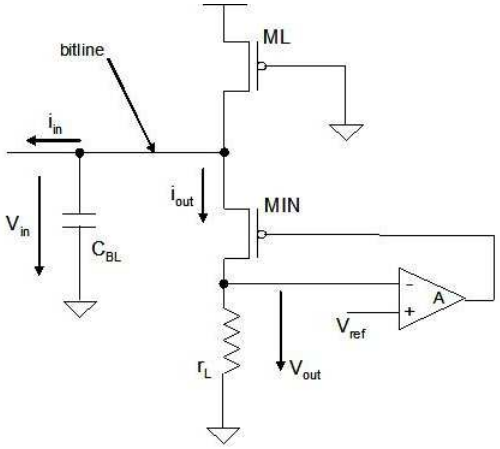


Figure 3.16: Type C Current Sensing

output voltage is fed back to the gate of the input transistor. As the input current increases the output voltage decreases and this further causes the output current to decrease, decreasing the output voltage even more (positive feedback). Differential circuits can be made using this principle as well and can thus be used to sense the bitlines. Another practical implementation is as shown in Figure 3.17. This circuit

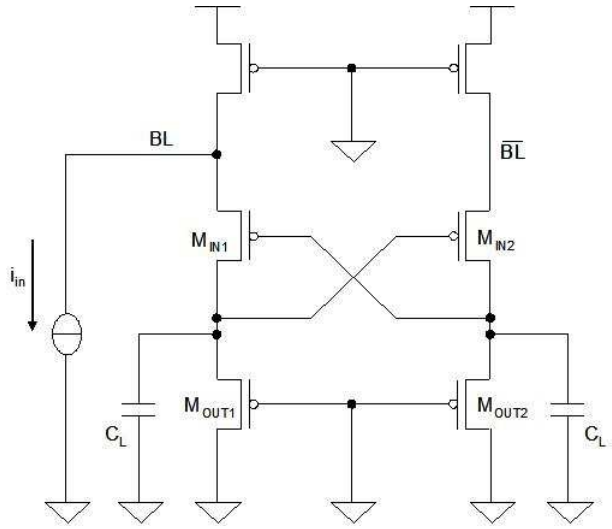


Figure 3.17: Seevinck's circuit[wicht35]

is considered as the circuit that proves the merit of current sensing in SRAMs. The major advantage of this circuit is its simplicity.

3.3.3.5 Type D - Wilson Type

Figure 3.18 shows a Type D current sense amplifier. It looks a lot like a Type B amplifier, but it has a complementary transistor and also uses a positive gain amplifier. By using the complementary transistor, $\acute{i}_{out}$ can be used along with v_{out}. This is also called a Wilson type amplifier since it resembles a CMOS Wilson Type current mirror [36]. A sense amplifier based on this principle is presented in [39].

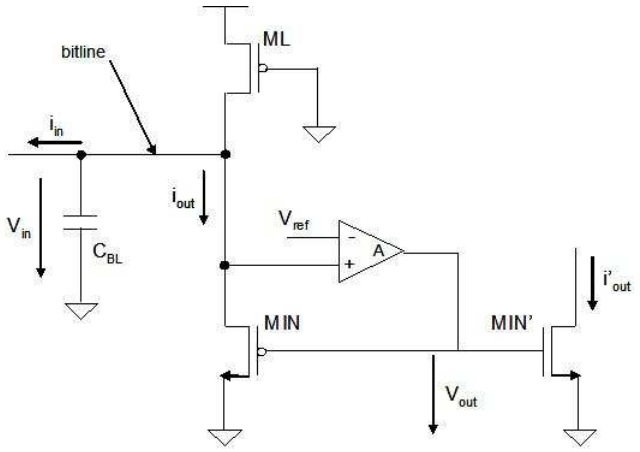
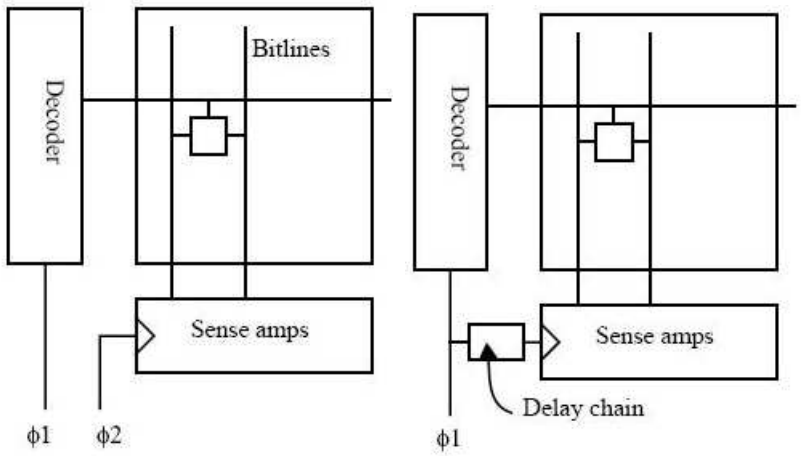


Figure 3.18: Type D Current Sensing

3.4 Circuit Techniques to generate Sense Enable Signal



(a) (b)

Figure 3.19: Common sense enable generation schemes

The most prevalent and simplest technique to generate the sense amplifier enable signal is to use an inverter chain. Two of the most commonly methods used are

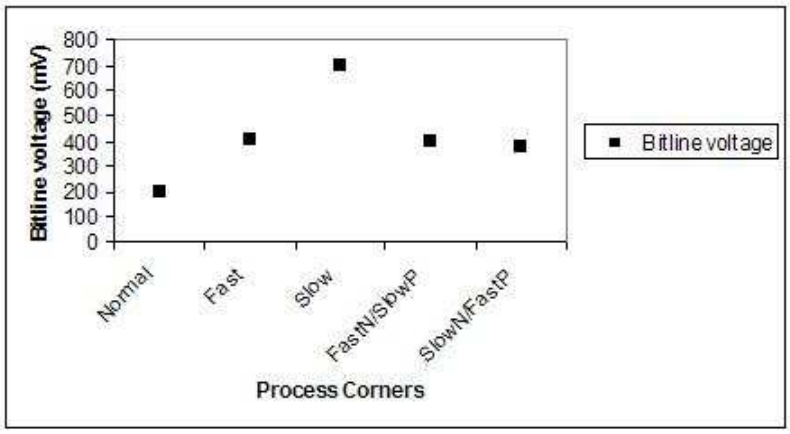


Figure 3.20: Variations in generation of sense amplifier enable using inverter chain scheme

shown in Figure 3.19. One uses the clock phase to do the timing (Figure 3.19(a)) while the other method uses a delay chain which is triggered by the block select line (Figure 3.19(b)). One of the major drawbacks of these schemes is that they do not track the delay of the memory cell over various processes and temperatures.

Figure 3.20 shows the variations in the generation of the sense amplifier enable output by using an inverter chain, over the various processes. Under nominal conditions the inverter chain was designed to fire the sense amplifier enable signal when the bitline differential voltage was 180mV. The inverter chain was designed so that the Sense Enable signal reaches 90% of it's final value (1.6V) just as the bitline differential voltage is 180mV. The process used was $0.18\mu m$ and a 256 by 16 bit memory array was used. As can be seen, there are wide variations over the various process corners. The primary reason being that while the memory cell delay is mainly affected by the nMOS thresholds, the inverter chain delay is affected by both nMOS and pMOS thresholds. The worst case matching for the inverter delay chain occurs for process corners where the nMOS and the pMOS thresholds move in the opposite direction. There are two more sources of variations that are not included in the graph above and make the inverter matching even worse. The minimum sized transistors used in memory cells are more vulnerable to delta-W variations than the non-minimum sized

devices used typically in the delay chain. We will next look at two methods explored by [2] to generate the enable signals, which track the bitline variations. Using these schemes we can also control the word line width, which minimizes power dissipation.

3.4.1 Replica delay based on capacitance ratioing

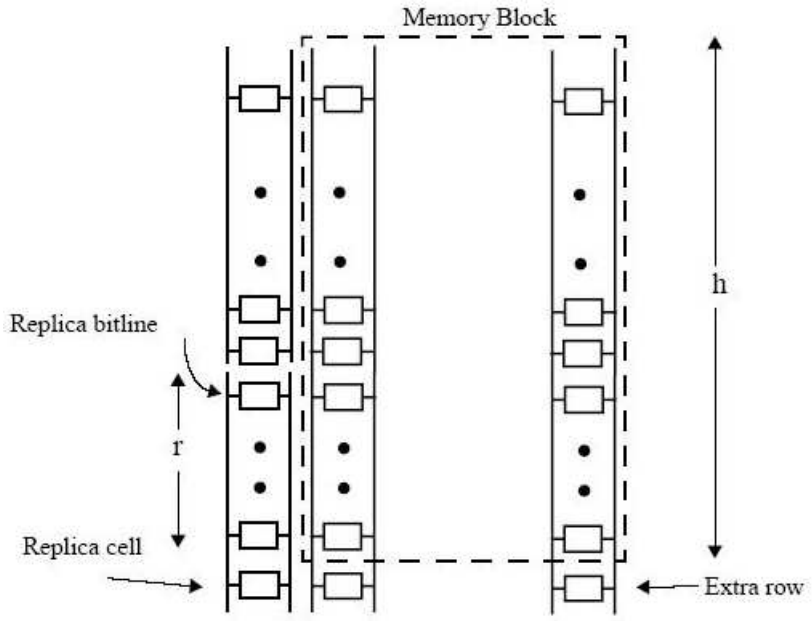


Figure 3.21: Replica Column with bitline capacitance ratioing[2]

One of the methods to control the variations due to process parameters is to use a replica delay element to generate the sense amplifier signal. As shown in [2] one way is to use a memory cell which drives a small section of a replica bitline. The short bitlines capacitance is set to be a fraction of the main bitline. The length of this short bitline, is determined by the bitline swing needed. Generally, one-tenth of the full swing is considered enough for the sense amplifier to read the correct value. Hence, if we using a 1.8V supply, the typical bitline swing required by the sense amplifier to work properly is 180mV. The length of the short bitline is thus made so that it triggers the sense amplifier when the bitline swing is 180mV. The replica bitline has a similar structure to the main bitlines in terms of the wire and diode

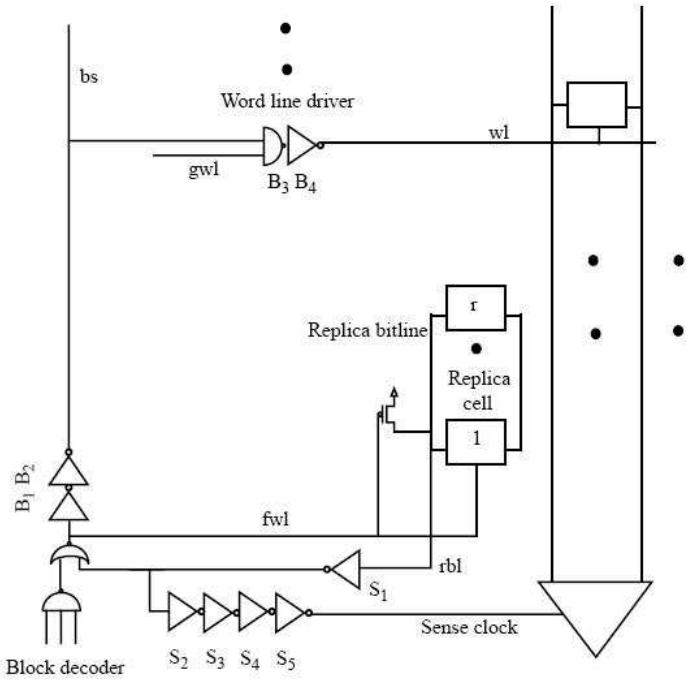


Figure 3.22: Working of Bitline Capacitance Ratioing Scheme[2]

parasitic capacitances Figure 3.21. Hence its capacitance ratio to the main bitlines is set purely by the ratio of the geometric lengths, r/h. The replica memory cell is programmed to always store a zero so that, when activated, it discharges the replica bitline.

Figure 3.22 shows the working of this circuit. The replica bitline is initially held high using the *fwl* signal. Thus, the output from the inverter S1 is low. When a block is selected, the output of the NOR gate goes high, thus enabling the replica memory cell. At the same time, the replica bitline is floating. The output from the NOR gate is buffered and ANDed with the global word line signal to select the corresponding word line. Simultaneously, the replica memory cell discharges the replica bitline, causing the output of S1 to go high. This disables the NOR gate and this causes the word line to be deselected as well. The output from the replica bitline, though a full swing signal, has a very slow slew rate and this needs to be buffered before it can drive the following stages. It is thus buffered by a chain of inverters S1-S5. If

the delay caused trough the chain S1-S5 can be made equal to the delay from B1-B4, then the replica bitline output can be made to fire the sense amplifier at the exact moment desired. We have used this scheme in our design to enable the sense amplifier to fire when the bitline swing is 180mV.

One of the disadvantages of this scheme is that even though it generates well controlled sense clock edges, the wordline pulse width turn out to be longer than the minimum desired. This causes the bitline voltages to swing more than the minimum required and leads to more power consumption. This is mainly because the word line reset signal has to travel back from the replica bitline through the B chain which is usually optimized to speed up the set signals to assert the local word line in order to reduce the access times. In the next section we will look at an alternative replica scheme [2] which allows for feeding the replica signal directly into the word line driver.

3.4.2 Replica delay element based on cell current ratioing

As shown in [2] this scheme uses an extra row and column. The extra row has memory cells, whose pMOS devices have been removed, thus making them current sources Figure 3.23. All their outputs are tied and they discharge the replica bitline simultaneously. Thus, the numbers of these memory cells determine the slew rate of the replica bitline discharge. These current sources are activated by the replica word line which is enabled during every access of the block. The replica bitline is identical to the main bitline, and thus has the same parasitic capacitances as the main bitline. The local word line are skewed to increase the speed of the rising transition, and they can be reset by the replica bitline as shown in Figure 3.24. The reset signal for the word line is connected to the access transistor as shown. Thus, when the replica bitline is discharged by the current sources, the signal F1 resets the word line. Thus, this scheme is better at controlling the word line pulse width as compared to the previous method, since the reset signal does not have to go through a chain of gates, but instead directly resets the word line driver.According to [2] this technique is faster by about 2/3rds of a gate delay as compared to the replica bitline scheme based on capacitance ratioing. However, this scheme takes up more area and also

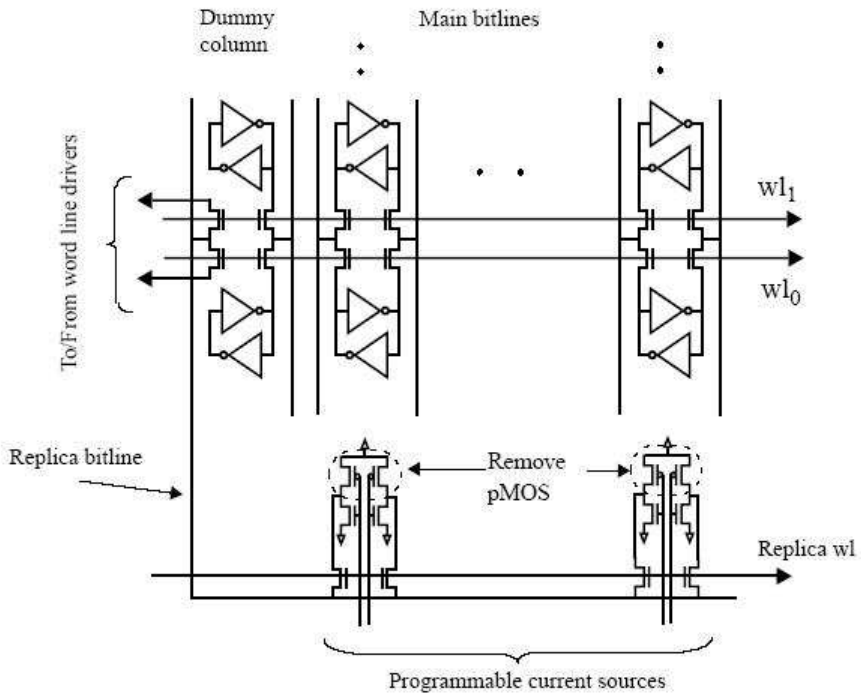


Figure 3.23: Current Ratio Based replica scheme[2]

consumes more power due to the current sources. Thus, this scheme is good for large access width SRAM's.

3.5 Compensation schemes for Bitline Multiplexer

As shown by [46, 43] so far, a main limitation of the practical use of current sense amplifiers is the finite resistance of the bitline multiplexer (MUX). It has been shown by [8] that the resistance of the bitline multiplexer can reduce speeds up to 5ns. [43] have shown that due to the resistance of the multiplexer the current signals decrease, the sensing delay increases and finally the merit of current sensing disappears. Voltage sensing designs also face the same difficulties. As shown in [21], a faster bitline sensing has been proposed by placing the multiplex behind the sense amplifier. [21] like many other publications propose to increase the width of the bitline multiplexer to improve the delay.

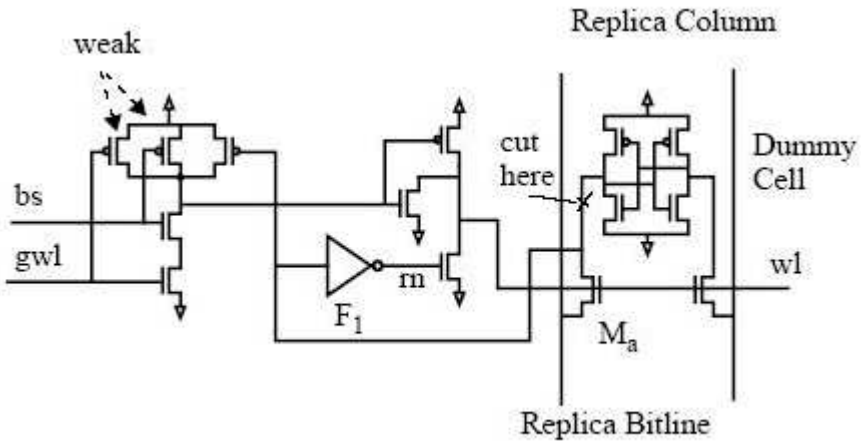


Figure 3.24: Skewed Word Line Driver[2]

3.5.1 General considerations for design of multiplexer

Figure 3.25 shows the different ways in which the multiplexer connects the bitline and the sense amplifier. Using both transistor types, the multiplexer and the sense amplifier can act as a current sink and a current source. The sense amplifier is modeled by the small signal input resistance r_s and by the DC voltage V_{ref} of the current sensing input stage. Since the potential is well defined, it is not necessary to use a transmission gate. As can be seen from the figure we have to make sure that the multiplexer transistor completely turns on ($|V_{GS}| > V_{th}$).

For the n-channel transistor of Figure 3.25(a), the transistors drain is the input terminal and hence the output characteristic of the transistor defines the effective input resistance. To get a low input resistance needs to be biased in the linear region. In the saturation region the input characteristics of the MOS transistor becomes nearly independent of the drain potential. Hence, the effective small signal input resistance in saturation is independent of r_s and the sense amplifier becomes ineffective. Therefore, when using current sense amplifier we have to make sure that the multiplexer transistor is in the linear region.

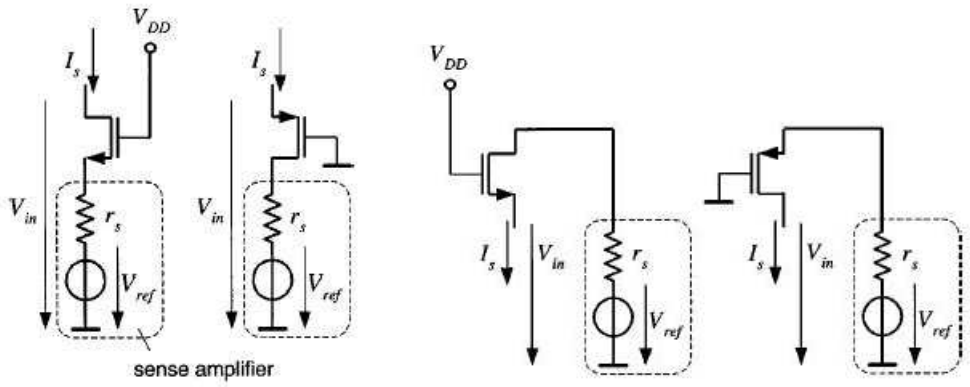


Figure 3.25: Multiplexer and Sense Amplifier operating as current sink (a) and current source (b).[8]

3.5.2 Techniques for Multiplexer Compensation

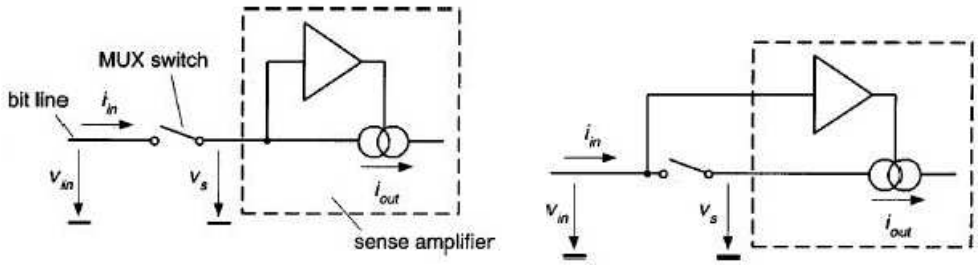


Figure 3.26: Idea of Multiplexer compensation.[8]

[8] have proposed a new technique for multiplexer compensation. It mainly applies to current sense amplifiers where the input voltage is measured and controlled by the internal feedback loop. Figure 3.26(a) shows the general principle of operation. The controlled current shown, refers to the MOS transistor which is present in the feedback loop. We can assume $v_{in} = v_s$ for an ideal multiplexer. For the multiplexer in 5.6a, the sense amplifier holds its input voltage vs constant, but not v_{in}. The multiplexer resistance thus adds to the input resistance of the sense amplifier. The impedance

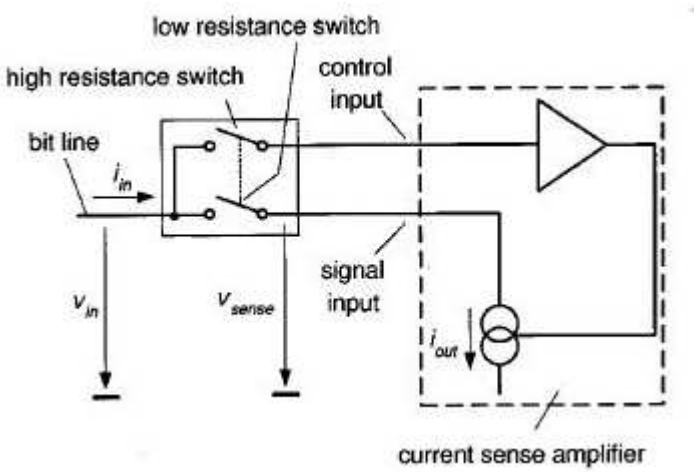


Figure 3.27: Proposed Circuit using a double multiplexer switch.[8]

at the bitline node increases and causes significant speed degradation as well as a decrease in the signal current.

The scheme proposed by [8] is as shown in Figure 3.26(b).It introduces a direct control of the voltage v_{in} before the multiplexer, i.e., the high capacitance bitline. A second switch needs to be added in order to maintain the multiplexer functionality. The new scheme is as shown in Figure 3.27. As can be seen from the figure, the input of the sense amplifier is now split into a signal and a control input. This two-switch concept decouples the signal detection and signal processing in a very efficient way. Both the signals are controlled by the same select signal. Since there is no current flowing in the control path, we can use a minimum size transistor as the additional switch. The scheme can also be expanded for n bitlines.

3.6 Low Power Writes

Generally the bitline is discharged all the way to ground during a write operation. This huge bitline swing can cause large power consumption during writes. During a read operation, the bitline voltage swing is generally restricted to around 200mV, and thus the writes can on an average consume about $1/8^{th}$ more power than a read

operation. One method to reduce write power consumption has been proposed in [19] and involves partitioning the bitline into very small segments with very small capacitances. Another method proposes using small voltage swings in the bitlines to do the writes. However, the key problem in achieving writes using a small bitline voltage swing is to actually overpower the memory cell using such a small swing.

Mai et. al [22] have proposed a technique where they bias the bitlines near the ground voltage. When a small bitline differential is set up for the write and the word line is activated, the internal nodes of the memory cell are quickly discharged to the values of the bitlines through the access nfets. When the word line is turned off, the small differential voltage between the cell internal nodes is amplified by the cross coupled pfets in the cell. To prevent the reads from over writing the cell, the authors propose to reduce the word line voltage for the read operation, thus weakening the access nfets and preventing a spurious discharge of the cell internal nodes. The main weakness of this approach is that read accesses become slower now since the word line voltage is smaller. Mori et. al. in [31] use a similar concept, except that they use a bitline reference of $V_{DD}/2$ which is easy to generate and incurs significantly lower read access penalty. A write is achieved by discharging one of the bitlines to ground. Since the write bitline swings are halved, a factor of four savings in the bitline write power is achieved. To improve the robustness of reads, the cell voltage is boosted above V_{DD}.

[1] have proposed a technique where they use small bitline swings to write the data, while maintaining the bitline reference close to V_{DD}. The small bitline swings are amplified by operating the memory cell as a latch sense amplifier as shown in Figure 3.28. This is achieved by modifying the ground connection to the memory cell.All the cells within a row share this common line called the virtual ground line. During a read operation, this line is pulled low and the memory cell behaves like a typical cell. However, before a write operation, this virtual ground line is pulled high, resetting the contents of the memory cells. The write data is put on the bitlines in the form of small voltage swings. The virtual ground line is then pulled low, causing the data to be latched by the cell which amplifies and stores this data. This approach has a few limitations though. Since boosted word lines have to be used, the access

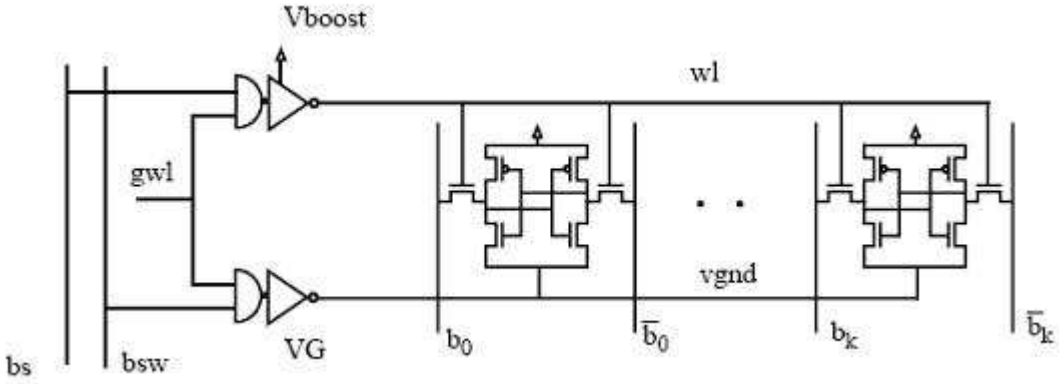


Figure 3.28: Scheme for Low Power Writes [1]

nfets have higher current drive which makes the cell less stable during reads. Also, all the cells in a particular row are reset at a time since they all share the virtual ground line. Thus, the block width needs to be the same as the write word width, else the data has to be first read out onto the bitlines, and then rewritten back into the cells.

Chapter 4

Design of 4K SRAM

In the previous chapter we looked at the various circuit techniques which can be used to make a fast low power SRAM. In this chapter we have described the design of a 4KB SRAM based on those techniques. We first describe the architecture that was chosen for our design. Next we describe the decoder design followed by the design of the memory cells and the sense amplifiers.

4.1 Design Architecture

As described in Chapter 2, there are three main architectures prevalent in SRAM design's today- 1) Monolithic architecture 2) DWL architecture 3) HWD architecture. Based on the size of the memory array a particular architecture is chosen. The DWL and the HWD architectures are generally preferred for larger designs, since they help in reducing the RC delay of long word lines. One of the drawbacks of these architectures is the added area, mainly in the form of word line drivers that have to be placed at the edges of each block or sub-block. As shown by Hirose et. al. [34] there are no significant advantages of using the HWD or the DWL architectures for smaller memory densities. The advantages can be seen for memory densities greater than 16kB.

Since our design is a 4kB design, there are no significant advantages in using either the DWL or the HWD architecture. In fact both these architectures would just

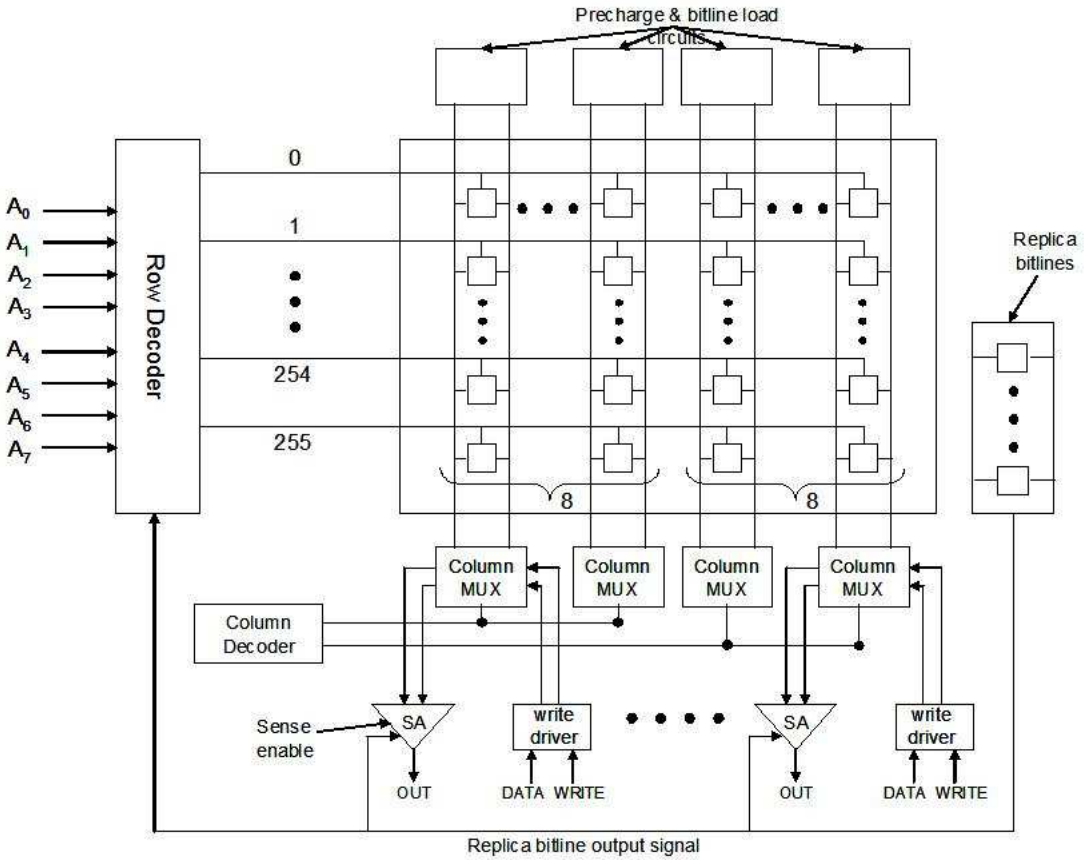


Figure 4.1: Block Diagram of 4K design

increase the area of the design, without really affecting the performance in terms of delay. Thus, for our 4K design we have chosen a monolithic architecture. Figure 4.1 shows the block diagram of our design. The memory has been designed as a 256 row by 16 column array of cells. A 8 to 256 decoder is used to select one of the 256 rows. The columns are split into two 8-bit wide sub-words. Thus, a byte of data can be written or read during each read/write cycle. Voltage sense amplifiers (alpha latch) have been used to read the data from the bitlines. The replica bitline architecture based on capacitance ratioing has been used to fire the sense amplifiers at the most optimum time. This output is also used to reset the word lines so as to reduce power consumption and minimize the bitline swings as described in Chapter 3.

4.2 Decoder Design

As described in Chapter 3, we have a lot of techniques available for designing a decoder to reduce both power consumption and area. As described by [29, 12, 10] pulsed decoders are preferred over the conventional combinational based decoders. Pulsed decoders are a lot faster than the combinational decoders, though at the cost of power. At any given time, only one decoder output can be selected, thus in combinational based decoders, we have to first disable a previously selected output, and then assert the new output. The total delay involves the delay in deselecting the previous output in addition to the delay in asserting the new output. In a pulsed decoder the decoder output is reset after a given time and thus the delay involved is the delay in asserting the new output only. Care must be taken to make sure that the word line pulse width is wide enough to enable a particular memory cell so that data can be read/written to it.

There are various schemes which can be used to reset the decoders. These generally include some sort of external reset chains, which are either driven by some external signals or by the inputs to the decoders themselves. Two such techniques - SRCMOS and DRCMOS have been described in chapter 3. These schemes however, are effective for the second and beyond decoding stages. One of the disadvantages of these schemes is the increase in area, due to the addition of the reset gates.

Our design only has a single decoding stage, an 8 to 256 decoder. Also, the capacitance being driven by the decoders is very small (around 30fF), since the memory array spans only 16 columns. The RC of the interconnect is also very small in the order of a few fF and around 30Ω. The layout has been designed so that the two blocks of 2K memory arrays are placed on either side of the decoding stage which is placed right in the middle. So the word line capacitance is further reduced by a factor of two.

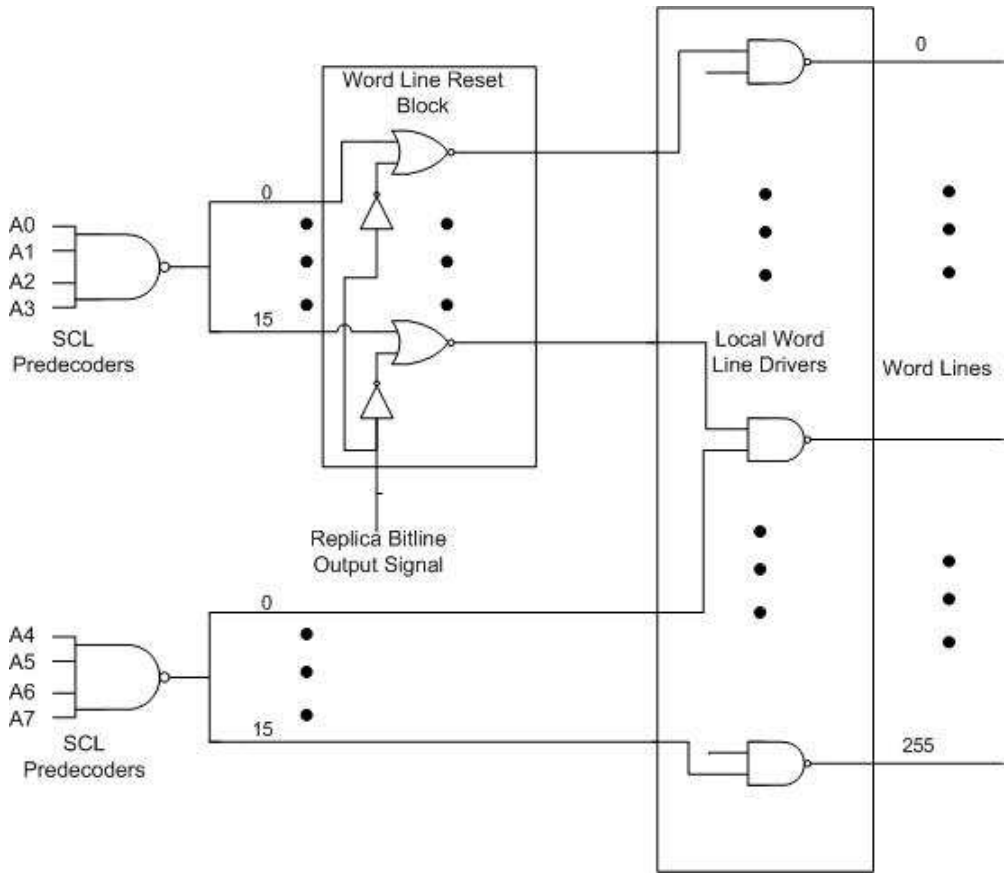


Figure 4.2: Block Diagram of Decoder Structure

4.2.1 Decoder Architecture

The decoder structure used for our design is as shown in Figure 4.2. It consists of two sets of 4 to 16 decoders, which decode the address line A0 - A7. The outputs from one set of the 4 to 16 decoders is driven through a Word Line Reset Block. The decoder outputs (which are active low), are given to one input of a NOR gate. The other input of the NOR gate comes from the replica bitline output (described in section 4.5.2). Initially the replica bitline output is low, so when a particular decoder output goes active (low), the NOR gate gets enabled and drives a 1 (V_{DD}) on its output line. This is given to one input of the local word line driver circuit. The other input of the local word line driver comes from the second set of 4 to 16 SCL decoders.

Thus, the outputs from both sets of SCL decoders are combined by the local word line drivers to obtain the 256 word line outputs.

The replica bitline circuit is used to minimize the word line pulse width while making sure that it does affect the functionality of the circuit. As described later on the replica bitline circuit is used to enable the sense amplifier at the most optimum time. This is also used to disable the local word line driver and helps reduce power dissipation. Assume that a zero is stored in a memory cell which has been selected. When the word line is driven by the local driver, this memory cell will get enabled. During the read operation, the bitline BL will start discharging while the bitline $\overline{BL}$ will remain at V_{DD}. Thus a differential voltage starts to develop between the two bitlines. If the word line is kept enabled, then this differential voltage will keep increasing till the bitline BL is completely discharged to zero. Once the read operation is complete this bitline has to be precharged again to V_{DD} for the next operation. This is very power intensive since the bitlines have a large capacitance which needs to be charged. By minimizing the word line pulse width we can minimize the bitline swing reducing power dissipation. The replica bitline circuit has been designed to enable the sense amplifiers when the bitline differential voltage is around 180mV, and this same signal can be used to disable the local word line drivers. Thus, once the replica bitline circuit fires, it disables the NOR gate which in turn turns the local word line driver off. Even though there is a lag between the replica bitline signal arriving and the local word line driver actually switching off, we are still able to reduce the word line pulse width and reduce the power dissipation. We ran simulations on the 8 to 256 decoder using this resetting technique. We compared this with a similar 8 to 256 decoder, without the replica bitline output signal. As shown in Table 4.1 we can see that there is a reduction in power dissipation when using this resetting technique.This is due to the fact, that the word line pulse width is reduced from 0.86ns to 0.67ns. Due to the reduction in the pulse width, the bitline swing is reduced, and this reduces the power dissipation in charging the bitlines after every read cycle.

	Power Dissipation	Word Line Pulse Width	Maximum Bit-line Swing
Decoder Structure using replica bitline scheme to reset word line	18.3mW	0.67ns	160mV
Decoder Structure without the replica bitline scheme	23.6mW	0.86ns	280mV

Table 4.1: Comparison of Decoding schemes with and without the replica bitline reset signal

4.2.2 Design of SCL Predecoder Circuits

We have designed a decoder using SCL circuits combined with reset circuits, as has been described by [12]. The decoder design is as shown in Figure 4.3.The figure shows the schematic used for the 4 to 16 decoder. This design has various advantages over the conventional NAND based design. At any point, for any combination of inputs, we only have one pMOS and two nMOS transistors in series. This reduces the delay involved. For a similar 4 to 16 decoder designed using the conventional NAND gate, we would have one pMOS and four nMOS transistors in series, which would increase the capacitance and add to the delay.

When the clock signal (clk) is at a low level, the drain nodes of the four nMOS's are precharged to a high level in the circuit (precharge phase). If at least one of the input signals of the circuit is at a high level and clk then turns to a high level, node N is discharged to a low level and node M remains at a high level (evaluate phase). On the other hand, if all the input signals are at a low level and clk then turns to a high level, node M is discharged and node N remains at a high level (evaluate phase). Thus, the SCL circuit operates as a NOR/OR gate. In the circuit, only one pMOS and two nMOS's are connected in series like dynamic CMOS circuits. As mentioned earlier, the decoder using the SCL circuits can achieve higher speed than that of a decoder with conventional CMOS circuits.

A fanout of 4 was considered when designing this circuit. The pMOS transistors are used only during the precharge phase of the operation and hence can be skewed.

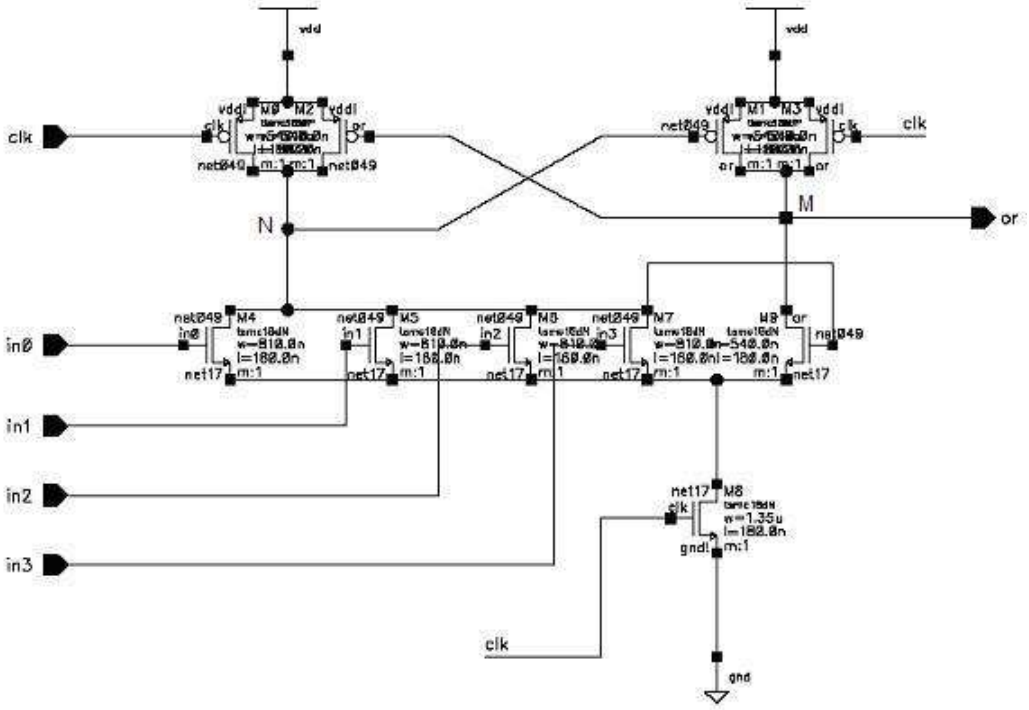


Figure 4.3: Decoder Design

For the decoders we are only concerned for signals that propagate in one direction. For this design we are concerned with any input being 1, since only for that condition will the decoder be activated. For that reason the pMOS transistors are skewed and made smaller. For our design we have chosen the pMOS size to be

$$pMOS\ width = 2 \times minimum\ W = 540\eta m. \tag{4.1}$$

One of the advantages of using a NOR style gate is that inputs can be added in parallel. The total resistance of the gate reduces since the resistances of these inputs come in parallel with each other. However, for the worst case, we have to consider that one only input is ON. The width of the nMOS gates was calculated based on the load capacitance that they need to drive. [16] and the work on Logical Effort [44] was used as a reference to determine the size of these transistors for least delay. The

word line load capacitance is very small, in the order of a few fF. Also, the first set of decoders has to drive a NOR gate (which is in the Word Line Reset Block) while the second set of decoders is used to drive the local word line drivers. As such the load on the decoders is very small. After calculations and running SPICE simulations, the optimum size for the input transistors was found to be

$$nMOS\ width = 3 \times minimum\ W = 810\eta m. \tag{4.2}$$

The nMOS transistor which is driven by the clock has to be made the largest. This is because it is at the bottom of the vertical stack. Its resistance thus needs to be reduced by increasing its size (progressive transistor sizing [16]). Also, we need to make sure that the clock signal does not get loaded. Hence, the size of the nMOS driven by the clock is chosen to be

$$nMOS\ width = 5 \times minimum\ W = 1350\eta m. \tag{4.3}$$

The decoder design is verified in the next chapter. We will next look at the design of the reset circuit.

4.2.3 Design of Reset Circuit

The reset circuit mainly consists of a NOR gate.As seen the pMOS transistors are made twice as big as the nMOS gates.This is because a pMOS transistor of width 2x delivers about the same current as a nMOS transistor of width x. One input to this NOR gate comes from the replica bitline signal, which is inherently a weak signal. Thus, we need to make sure that these transistors are large enough so that they don't load the replica bitline signal. Also, as shown in the previous section the SCL decoders are not made too big and in fact are around 3 times the minimum width. We have designed the circuits keeping a fanout of around 4. Hence, the pMOS transistors are made 10 times the minimum width while the nMOS transistors are made half of that.

$$pMOS\ width = 10 \times minimum\ W = 2.7\mu m \tag{4.4}$$

$$nMOS\ width = 5 \times minimum\ W = 1.35\mu m \tag{4.5}$$

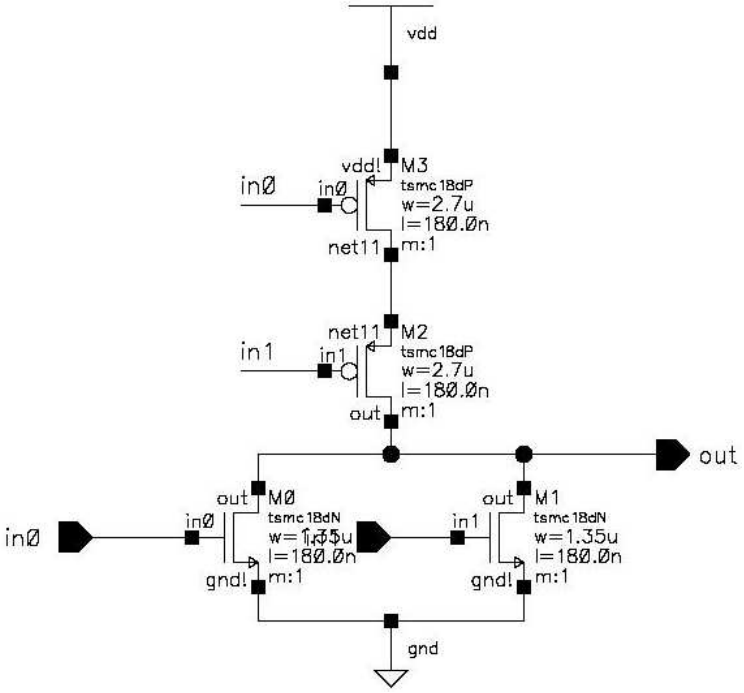


Figure 4.4: NOR gate used in reset block

4.2.4 Design of Local Word Line Drivers

The two 4 to 16 decoders are combined by using the local word line drivers Figure 4.5.The output from the reset circuit (Word Line Reset Block) is given to the in_gate input of the local driver.When a particular local word line driver is selected this signal goes high and enables the nMOS pass transistor and disables the pMOS. The in_drain inputs come from the other set of 4 to 16 decoders, so for a given address, one of those decoders will become active and pull the in_drain input low. So, for a particular local word line driver to be "ON", the in_gate input has to go high and the in_drain input has to go low. This causes a 1 at the output of the driver and drives the word line of the selected row. Once the minimum word line pulse width is achieved, the replica bitline signal deactivates the reset circuits, which in turn pulls the in_gate low. This switches the nMOS off and switches the pMOS on, thus disabling the word line for the row.

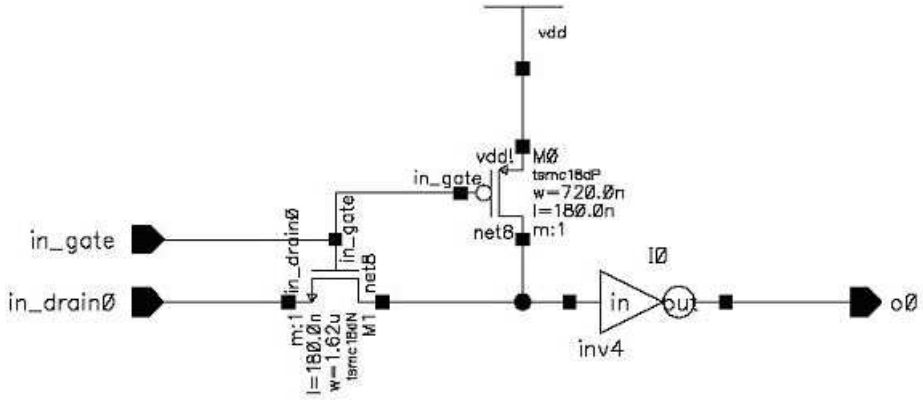


Figure 4.5: Local Word Line Driver

The emphasis is again on the signal propagating in the forward direction, i.e. the signal to turn the row word line on. Thus, the nMOS is made larger than the pMOS, since it is the nMOS which controls the output to the word line. The sizes for the nMOS and pMOS are as shown

$$nMOS\ width = 6 \times minimum\ W = 1.62\mu m \tag{4.6}$$

$$pMOS\ width = 2.5 \times minimum\ W = 720\eta m \tag{4.7}$$

An inverter buffer is used to drive the actual word lines. The nMOS for the inverter is made 4 times the minimum width i.e. $810\eta m$ while the pMOS of the inverter is made 8 times the minimum width i.e. $2.04\mu m$.

4.2.5 Conclusion

A simple 8 to 256 decoder was used to drive the word lines. A unique resetting technique to reset the decoders was employed which reduced the power dissipation by reducing the bitline swing, though for a small increase in area. Simple analytical techniques combined with extensive SPICE simulations were used to find the optimum transistor sizes. The design was tested extensively , across all process corners to make sure that it works. These results have been summarized in the next Chapter 5.

4.3 Memory Cell Design

As was described in Chapter 3 , the 6T and the 4T memory cells are the most prevalent in the industry. [9] have shown that for low voltages, especially below 2V, the 6T based cell is a lot more stable than the 4T cell. The 4T cells have a smaller area, but have a lower Static Noise Margin and are also more expensive to fabricate. Hence, for our design we have considered the 6T cells.

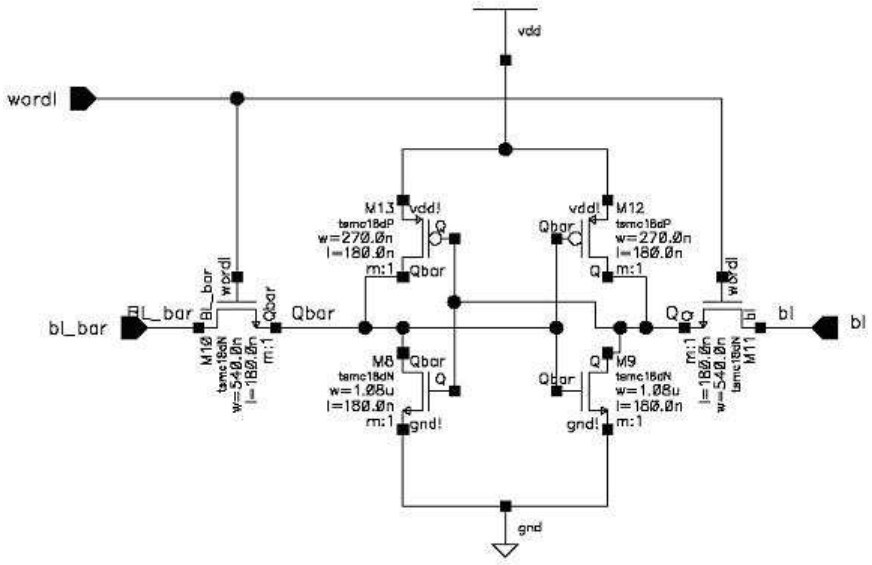


Figure 4.6: 6 Transistor Memory Cell

The schematic and the transistor sizes used are as shown in Figure 4.6.Transistors M8, M9, M12 and M13 form the cross coupled inverters while M10 and M11 are the access transistors.The cross coupled inverters store the actual data while the access transistors are used to enable the memory cell to either read/write data to the cell.

4.3.1 Design Considerations For Memory Cells

The various issues that have to be considered when designing the cell include - 1) Minimizing the cell area 2) Obtaining good read and write cell margins 3) Having good soft error immunity and 4) Having a good cell read current.Since the cell is

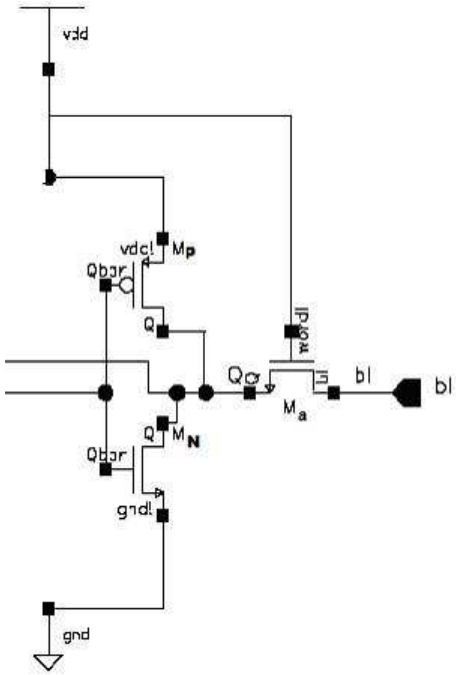


Figure 4.7: 3 Transistor Design Model

symmetric, we consider only half the cell (Figure 4.7) and design the three transistor sizes.The area of the cell is also an important criteria since we need to make sure that the cells can be as densely packed as possible.

4.3.2 Design for Read operation

Figure 4.8 shows how data is read from the cell. The bitlines are precharged and the word line of the cell is selected. As shown node 'a' contains a 0 bit while node $\overline{a}$ contains a 1. When the access transistors M3 and M4 are enabled, cell current I_{cell} flows from the bitline through node 'a' and down to ground. This increases the voltage at node 'a'. We need to make sure that the voltage at node 'a' does not become large enough so as to turn M2 on, because if M2 turns ON, then it reduces the potential at node $\overline{a}$ which can turn on M5 and cause the state of the cell to flip. This would then corrupt the data that is stored in the cell. It is important to make sure that the threshold voltage is the maximum allowable voltage at the internal node

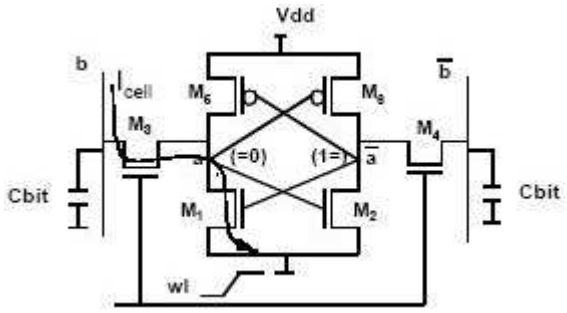


Figure 4.8: Reading data from a cell

during reads. As a rule of thumb,

$$W_N/W_a \sim 1.5 \; to \; 2 \tag{4.8}$$

4.3.3 Design for Write operation

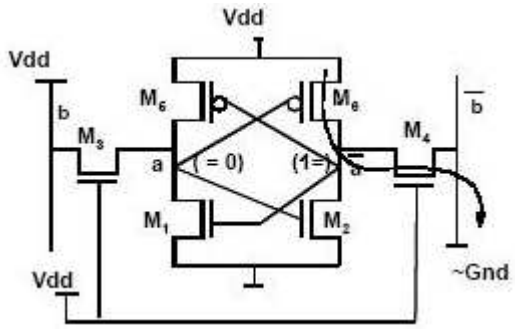


Figure 4.9: Writing data to a cell

Figure 4.9 shows how data is written into the cell. The data and the $\overline{data}$ is placed on the bitlines by pulling the appropriate bitline to V_{DD} or G_{nd}. We can assume that bit 0 is stored at node 'a' while bit 1 is stored at node $\overline{a}$. In order to write a 1 at node 'a', we place V_{DD} on the bitline while we pull $\overline{bitline}$ down to G_{nd}. The access transistors M3 and M4 are then selected. To store a 1 at node 'a', we need to pull node $\overline{a}$ low, so that transistor M5 gets switched on. Thus, transistor M4 needs to be

strong enough to pull node $\overline{a}$ low while transistor M6 is trying to pull this node high. We use the switching threshold as the trigger point for regenerative action to take place. As a rule of thumb, we choose

$$W_a/W_P \sim 1.5 \; to \; 2 \tag{4.9}$$

4.3.4 Transistor Sizing of memory cell

In order to achieve minimum area, we have to make sure that the transistors have minimum size. Also, from a SER point of view we need to size the transistors so that a read operation does not destroy the data and that a correct value is written into the cell during a write operation.Hence, we have chosen $W_a/W_P = 2$ and $W_N/W_a = 2$ so that we have good soft error immunity as well as good read and write margins. By choosing W_P as a minimum size transistor, we can obtain the least cell area. Thus we chose

$$W_P = 270\eta m \tag{4.10}$$

Accordingly the other two sizes come out to be:

$$W_a = 2 \times W_P = 540\eta m \tag{4.11}$$

$$W_N = 2 \times W_a = 1.08\mu m \tag{4.12}$$

Simulations were then run using the above transistor sizes, to make sure that the memory cell behaves as designed. These results have been summarized in Chapter 5.

4.4 Sense Amplifier Design

In chapter 3 we have looked at the current and voltage sense amplifiers. We also looked at practical implementations of some of these circuits. We will next compare the designs and show why we finally chose the voltage sense amplifier (Alpha Latch).

4.4.1 Case Studies

In this section we will look at case studies of two voltage sense amplifiers and one current sense amplifier.The alpha latch voltage sense amplifier and the latch-based cross coupled amplifier are the voltage sense amplifiers studied and Clamped Bitline Sense Amplifier is the current sense amplifier studied.

4.4.1.1 Alpha Latch

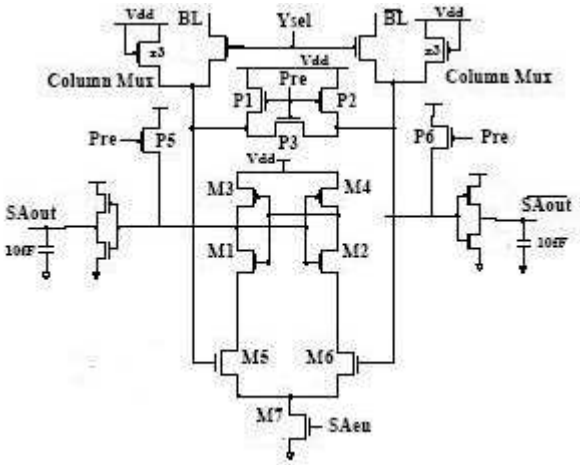


Figure 4.10: Schematic of an Alpha Latch Voltage Sense Amplifier

The alpha voltage latch [7] is shown in Figure 4.10. The alpha latch is similar to the standard Cross-Coupled Inverter latch [32], except that the bit-lines are terminated at the gates of two NMOS input devices and not the output nodes of the sense amplifier. The sense amplifier is fired when a sufficient bit-line differential is developed. To maintain it in the transient region before the sense amplifier is actually fired, the sense amplifier nodes are precharged and equalized. The advantage of this sense amplifier is that the bitlines are away from the sense amplifier output nodes and therefore provides lower swing in the bitlines.The precharge input makes sure that the amplifier nodes are precharged and equalized. The precharge is disabled at the same instant as the amplifier is enabled. The 'ysel' input is enabled when the data is ready to be read and is used to connect the amplifier to the bitlines.

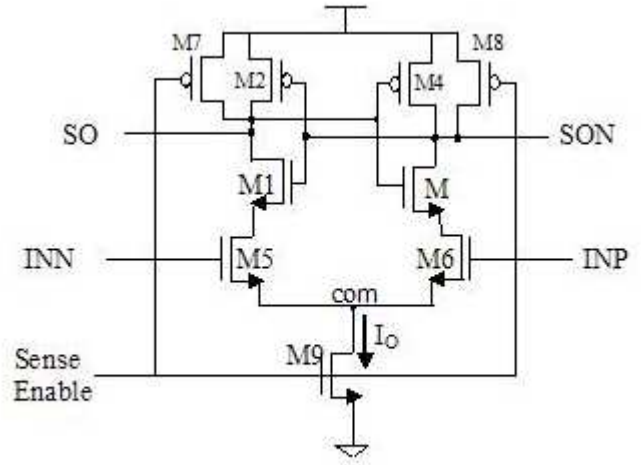


Figure 4.11: Latch-type voltage sense amplifier [30]

4.4.1.2 Latch-type voltage sense amplifier

The Figure 4.11 shows a latch type voltage amplifier [30]. This sense amplifier has strong positive feedback along with high resistive input. The current flow of the differential input transistors M5 and M6 controls the serially connected latch circuit. The small difference in the currents through M5 and M6 gets converted to a large output voltage. Since the Sense Enable line is low when the sense amplifier is disabled, the outputs are pulled up to V_{DD} by transistors M7 and M8. The sense amplifier is enabled by enabling the Sense Enable line which turns M9 on causing a current I_o to flow through it and brings the drains of M5 and M6 to be ground. Simultaneously, the transistors M5 and M6 are enabled. This causes the output nodes to be discharged. Since there is a small voltage difference between the inputs, the drain currents of M5 and M6 are also different. This causes different discharging speed at the outputs. Both the pMOS transistors now turn on when the output voltage reaches $V_{DD} - V_{thp}$. Strong positive feedback enhances the output voltage and causes one of the nMOS transistors to shut off. This amplifier is used very often in voltage sensed memories [40] as an alternative to the conventional latch. It is also used as an excellent second

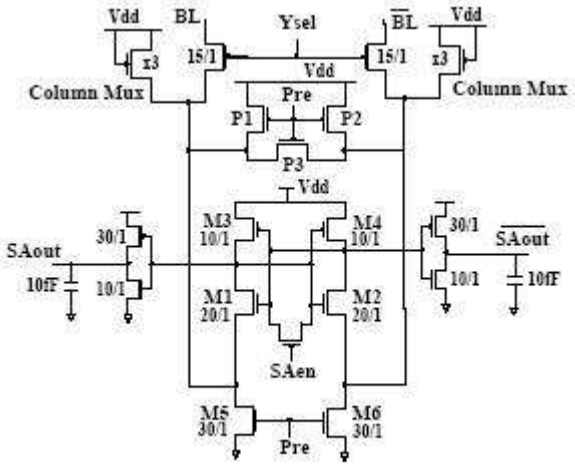


Figure 4.12: Clamped Bitline Sense Amplifier

stage after a current sensing input stage.

4.4.1.3 Clamped Bitline Sense Amplifier

Figure 4.12 [33] shows the schematic of a Clamped Bitline sense amplifier. The transistors M1-M4 form the cross coupled inverter pair, while M5 and M6 provide the low impedance termination. Transistor M7 equalizes the output nodes during the precharge phase. The output nodes are pulled high during the precharge phase. The transistors M5 and M6 are turned off by the precharge input. This prevents any static current during this phase, thus reducing the power consumption. The transistors M5 and M6 are turned on during the read operation, and when a sufficient differential is developed M7 is turned off. The cross-coupled inverter pair of M1-M4 now act as a high gain positive feedback amplifier.

4.4.1.4 Results

Simulations were performed using the Alpha latch voltage sense amplifier, the Latch based voltage sense amplifier and the Clamped Bitline current sense amplifier. Though the current sense amplifier was found to be faster than the voltage sense

amplifiers, the major disadvantage was the power consumption.The simulation setup consisted of bitlines spanning 64, 128 and 256 memory cells. Data was written into the first memory cell, and consequently read by the sense amplifiers. The amplifiers were

Circuit	Bitline Column of 64 memory cells		Bitline Column of 128 memory cells		Bitline Column of 256 memory cells	
	Delay	Power	Delay	Power	Delay	Power
Alpha latch voltage sense amplifier	234ps	0.22mW	240ps	0.23mW	256ps	0.25mW
Latch-type voltage sense amplifier	210ps	0.32mW	225ps	0.35mW	246ps	0.38mW
Clamped Bitline sense amplifier	196ps	2.47mW	215ps	6.6mW	227ps	12.2mW

Table 4.2: Comparison of Sense Amplifiers based on delay through memory cells

fired when the bitline differential voltage was 180mV. Table 4.2 shows the comparison of the three amplifiers in terms of the delay involved. The delay measured is from the word line reaching 10% of its final value (0.18V) to the output reaching 90% (1.62V) of its final value.As can be seen from the table the clamped bitline sense amplifier is the fastest. However, as can be seen from the Table 4.2 this comes at the price of power dissipation. The clamped bitline sense amplifier consumes a lot more power than the voltage sense amplifiers. Also, as seen in Fig[wicht 4.41], and shown by [46], there exists a delay crossover between current and voltage sensing. For small very small memories voltage sensing is preferred.

In our design we are using a 256 row bitline and the bitline capacitance is very small and hence we have considered the voltage sense amplifiers in our design. The latch-type voltage sense amplifier is faster than the alpha latch amplifier, but it also consumes more power than the alpha latch. The alpha latch was found to be more robust than the latch-type amplifier and was less prone to errors. This was especially true in cases when the sense amplifier was fired earlier than the optimum value, and hence there wasn't a sufficient voltage differential between the bitlines. As will be shown in the following sections, process variations affect the time at which the sense amplifiers are fired and the latch-type amplifier was thus more susceptible to errors.

4.4.2 Design of Alpha Latch Voltage Sense Amplifier

Some of the criteria that are considered when designing the alpha latch were having - 1) Good noise immunity 2) Minimizing the area 3) Having good stability. The area of the circuit had to be small, so that the final layout would fit within the pitch width of the bitlines.

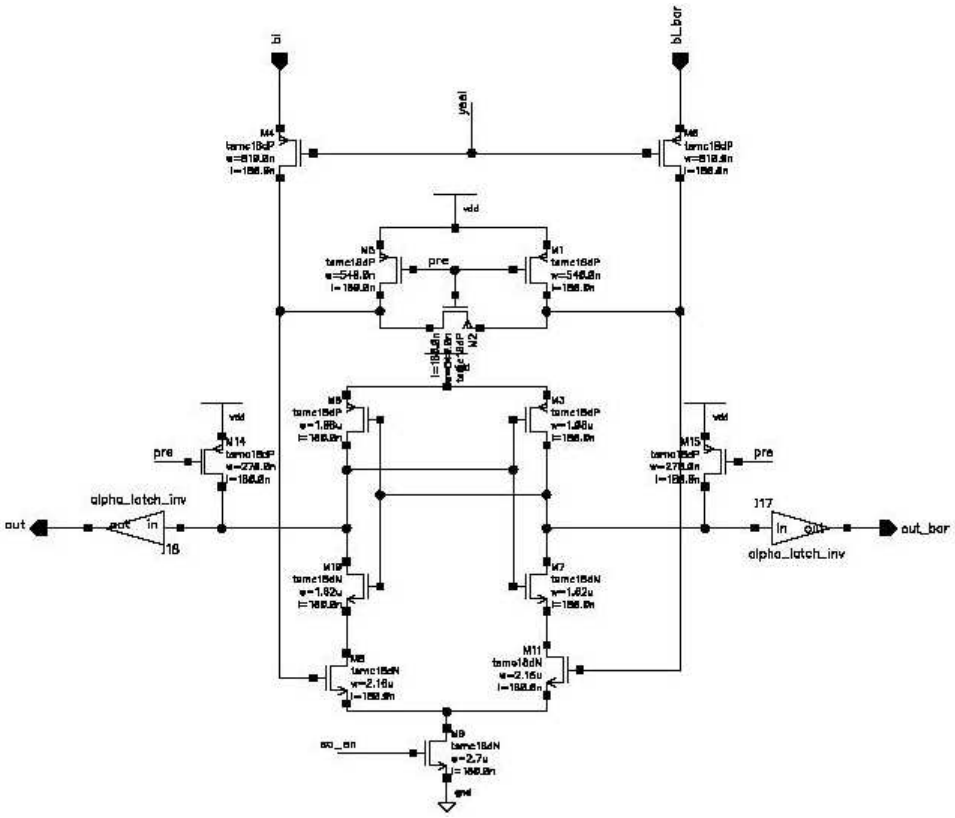


Figure 4.13: Schematic of Alpha Latch Voltage Sense Amplifier

As can be seen from Figure 4.13 the bit-lines are terminated at the gates of two NMOS input devices and not the output nodes of the sense amplifier. To maintain it in the transient region before the sense amplifier is actually fired, the sense amplifier nodes are precharged and equalized. The advantage of this sense amplifier is that the bitlines are away from the sense amplifier output nodes and therefore provides lower swing in the bitlines. The working of this amplifier can be explained with the help of

Figure 4.10. P1, P2 and P3 form the precharge circuits that keep the nodes at V_{DD}. P3 is used to equalize the nodes of the amplifier before it is enabled. P5 and P6 keep the output nodes precharged before the amplifier is enabled. These transistors can be minimum sized, since they don't have to drive a large capacitance and are switched off when the amplifier is fired. M1 - M4 form the cross coupled inverter pair which actually amplifies the signal to CMOS levels. M1 and M2 are made about 1.5 to 2 times the size of M3 and M4. M5 and M6 have their gate nodes connected to the bitlines, and hence see a large capacitance. They are made about the same size as M1 and M2. M7 is used to enable the sense amplifier and is thus made the largest to reduce its resistance.

4.4.2.1 Transistor Sizing of Alpha Latch

One of the major criteria when designing the sense amplifier was to make sure that its layout fits within the pitch of the memory cell. Hence transistor sizes were affected by this and this was kept in mind when choosing the transistor sizes. The structure of the sense amplifier is symmetrical and so we have to design sizes for M1,M3,M5 and M7 along with the precharge and equalization circuits (Figure 4.10). The precharge and equalization circuits are used to make sure that the sense amplifier is in the transient region before it is enabled.Thus, they have to keep the sense amplifier nodes precharged and are switched off when the sense amplifier is actually enabled. The capacitance that they have to drive is very small and so they can be made smaller.Based on SPICE results they are sized as twice the minimum width. Thus the sizes of P1,P2 and P3 are

$$P1, P2, P3 = 2 \times minimum\ W = 540\eta m \tag{4.13}$$

The precharge circuits keep the output nodes high and are disabled before the sense amplifier is enabled. They thus see a very small capacitance and are hence chosen to be of minimum size.

$$P5, P6 = minimum\ W = 270\eta m \tag{4.14}$$

When the sense amplifier is enabled, we can see that there is a vertical stack of nMOS transistors that is formed.These transistors are designed using the principle of pro-

gressive transistor sizing [16] so that the resistance of the transistors is progressively decreased. This approach reduces the dominant resistance, while keeping the increase in capacitance within bounds. From the Figure 4.10 we can see that the nMOS M7 which actually enables the sense amp is the last transistor in the vertical stack. This particular transistor is also driven by the replica bitline signal which is inherently a weak signal. Thus, this transistor is made the largest, so that it has the least resistance.

$$M7 = 10 \times minimum\ W = 2.7\mu m \tag{4.15}$$

The transistors M5 and M6 help isolate the output nodes from the bitlines. However this tends to slow the operation of the sense amplifier. In order to make the amplifier faster, these transistors have to be made wide, while at the same time we need to make sure that it is not too wide else it would lead to self-loading. Keeping the principle of progressive transistor sizing these transistors are made 8 times the minimum width of the transistors.

$$M5, M6 = 8 \times minimum\ W = 2.16\mu m \tag{4.16}$$

The cross-coupled inverters have to be designed so that they reach a stable point quickly, based on the bitline voltages.[42] have shown that the pMOS transistors should be made around 1.5 times smaller than the nMOS transistors. The nMOS transistors (M1 and M2) are made 6 times the minimum width,keeping in mind the principle of progressive transistor sizing. Accordingly the pMOS transistors (M3 and M4) are made 4 times the minimum width.

$$M1, M2 = 6 \times minimum\ W = 1.62\mu m \tag{4.17}$$

$$M2, M3 = 4 \times minimumW = 1.08\mu m \tag{4.18}$$

The transistors driven by the 'ysel' signal form part of the column multiplexer. According to [42] they should be made half the size of M7. This however makes the area of the sense amplifier larger than the pitch of the memory cell. Hence keeping that in mind, the size of this transistor is reduced. From SPICE simulations it is confirmed that it does not have an adverse effect on the working of the design.

$$nMOS\ Multiplexer = 3 \times minimum\ W = 810\eta m \tag{4.19}$$

4.4.3 Conclusion

After conducting case studies for both current and voltage sense amplifier, the voltage sense amplifier was chosen for our design. The Alpha Latch voltage sense amplifier was designed so that it can detect and amplify the differential voltages present on the bitlines. The amplifier was sized keeping in mind the pitch of the memory cells, and the design has been verified across all 4 process corners. The results of the tests have been given in Chapter 5.

4.5 Design for generation of Sense Enable signal

As shown in chapter 3 , we have three different schemes that can be used to generate the sense amplifier enable signal (Sense Enable). Of these, the inverter chain scheme is the simplest and has the least area. However, one of the major problems with this scheme is that it does not track the bitline over the various process variations. The other two schemes - The replica bitline based on capacitance ratioing scheme and the replica bitline based on current ratioing schemes track the bitline a lot better over the process variations. This, however comes at an added cost of area. The Replica bitline scheme based on capacitance ratioing adds an extra column, while the scheme based on current ratioing adds an extra row and column. Also, the scheme based on current ratioing consumes more power, since it has current sources, which sink a lot of current. This scheme has advantages for large density designs, and so for our 4K design it is just too area and power intensive.

4.5.1 Case Studies

In this section we compare the inverter chain scheme versus the scheme based on capacitance ratioing. For the purpose of simulation, we used a 256 row by 16 column memory array. A 8 to 256 decoder selected the appropriate row, while a 4 to 16 column decoder was used to selset the appropriate column. The Alpha Latch voltage sense amplifier was used to amplify the voltages on the bitlines.

4.5.1.1 Inverter Chain Scheme

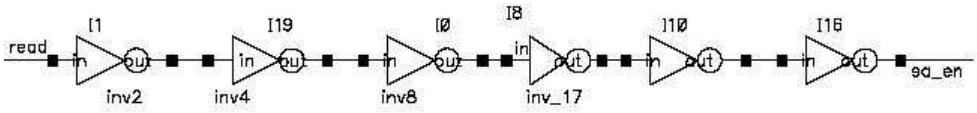


Figure 4.14: Inverter Chain used for sense enable signal

Figure 4.14 shows the schematic of the inverter chain that was used to generate the sense enable signal.As shown the inverters were made progressively larger with each stage. The size of the chain was determined by the delay needed to generate the sense enable signal, when the bitline voltage is close to 180mV. Thus, the sense enable output reaches 90% (1.62V) of its final value when the differential voltage is close to 180mV. Figure 4.15 shows the waveforms that were captured from awaves

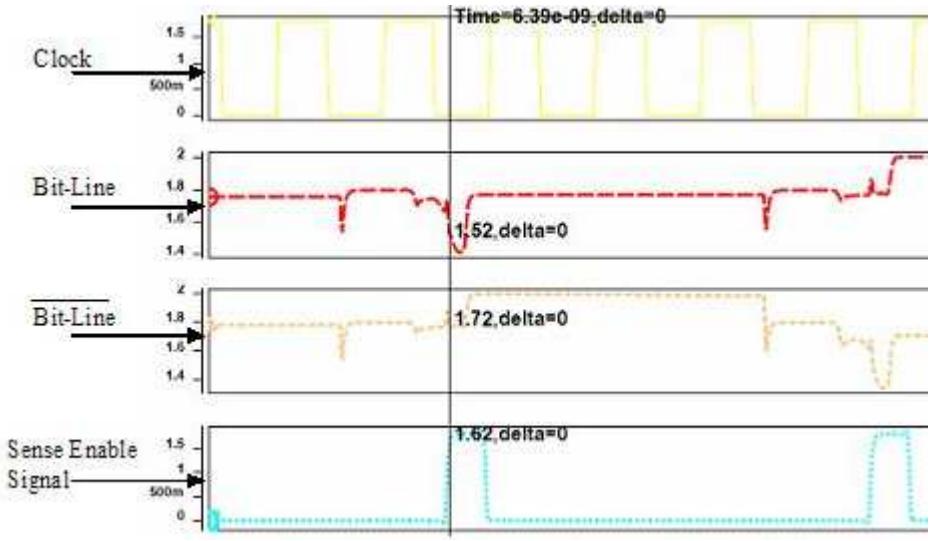


Figure 4.15: Waveforms showing timing scheme used in inverter chain scheme

after running the simulations. As seen, the sense enable signal switches on when the differential voltage between the bitlines is 200mV. The above simulations were also performed across the process corners and these results are summarized in section

4.5.1.3.

4.5.1.2 Replica Bitline scheme based on capacitance ratioing

The working of this scheme has been described in Chapter 3. The size of the replica bitline can be determined by the point at which we want the sense enable signal to be generated. The bitline length spans 256 rows while the full swing voltage is 1.8V. We need the sense enable signal to be asserted when the bitline voltage is 180mV which is 1/10th of the full swing signal. Thus, in order to get the sense enable signal at that point, we need to size the replica bitline to be about 1/10th the length of the bitline (i.e. 26 rows). After running some SPICE simulations to fine tune the design, so that it asserts the sense enable signal when the bitline voltage is 180mV, we chose the replica bitline to span 29 rows. Figure 4.16 shows the replica bitline

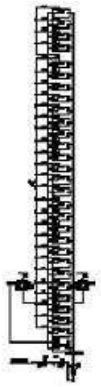


Figure 4.16: Replica Bitline Schematic

that was used for the simulations. From the working described in Chapter 3 we know that a dummy cell which permanently stores a 0 is needed. Certain delay elements are needed to ensure that the word line for this dummy cell is enabled simultaneously as the word line for the memory cells is enabled.This ensures that the sense enable signal is asserted at exactly the right moment as can be seen from the waveforms in Figure 4.17.As shown the sense enable signal reaches 90% of its final value when the bitline voltage is 180mV (1.78V - 1.6V).

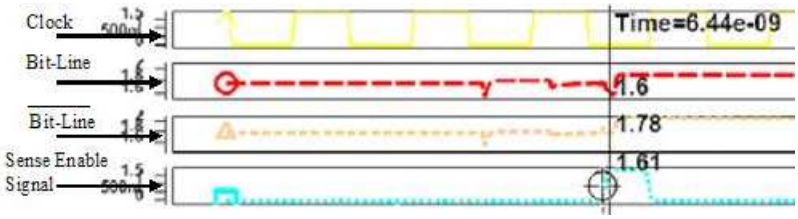


Figure 4.17: Waveforms for Replica Bitline Scheme

4.5.1.3 Results

Both the designs were setup so that the sense enable signal would be asserted when the bitline voltage is 180mV. Both designs were then simulated across the four process corners, to see how each design fared. Figure 4.18 shows the graph of the

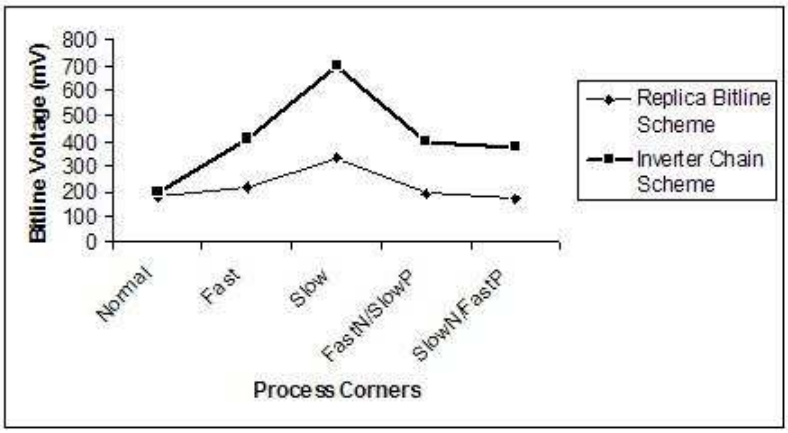


Figure 4.18: Replica Bitline Scheme vs Inverter Chain Scheme

results that were obtained from the simulations. The output from the inverter chain scheme varies a lot across the process corners, while the output from the replica bitline scheme seems to track the bitlines a lot better over the process corners. Thus the replica bitline scheme is definitely superior to the inverter chain scheme and is chosen for our design.

4.5.2 Design of Replica Bitline Scheme based on Capacitance Ratioing

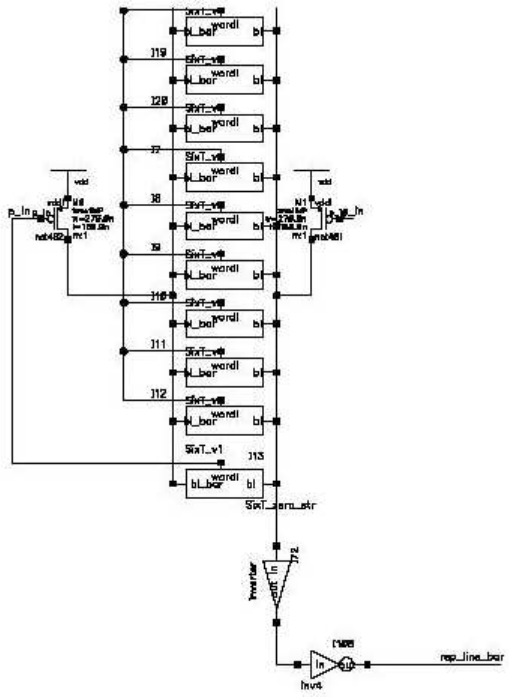


Figure 4.19: Working of Replica Bitline Scheme

From the previous section is it clear that the replica bitline scheme is definitely better than the inverter chain scheme and tracks the bitlines over the various process corners. In our design we are using a 256 row memory array, similar to the one used in the simulations. Hence the design of the scheme is very similar to the one mentioned in the previous section. We have used 29 dummy cells in the replica bitline structure. The output from this circuit is used not only to enable the sense amplifiers but to also disable the word line as explained in the previous sections. This signal is given to the Reset Word Line block in the decoder structure and when this output goes high, the word line is de-asserted. This signal thus needs to be buffered, though we cannot use too many buffers as this will increase the word line pulse width and also cause a delay in the firing of the sense amplifier, both of which increase the power dissipation. The

replica bitlines also need to be precharged until the dummy cell is selected, which then discharges the bitlines and enables the sense amplifier.Figure 4.19 shows the general setup that is used to precharge the replica bitlines. As seen two minimum size pMOS transistors are used to precharge the replica bitlines. These are enabled as long as the dummy cell word line is low. The word line for the dummy cell is activated when a particular word line is decoded and asserted by the row decoders.Simultaneously the pMOS transistors precharging the replica bitlines are switched off. Thus, the replica bitlines are left floating and since the dummy cell is programmed to store a 0 at all times, the bitline discharges. Thus, the rep_line_bar output goes from high to low. This is then buffered and given to the sense amplifiers and the Reset Word Line block.

4.5.2.1 Design of dummy memory cells

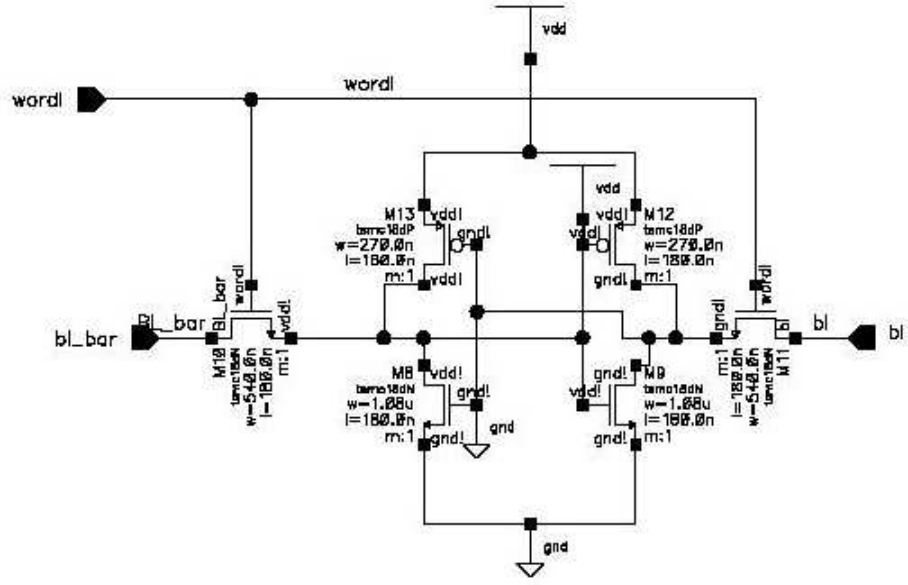


Figure 4.20: Dummy Memory Cell

The purpose of the dummy cells is to provide the same capacitance as the normal memory cells. Hence the transistor sizes are the same as those used in the design of the memory cells. However, the dummy cell which is accessed in the replica bitline

structure needs to store a 0 at all times. Hence its design is slightly altered as shown in Figure 4.20. The node connected to bitline is set to ground while the node connected to the $\overline{bitline}$ is set to V_{DD}. Thus, whenever this particular dummy cell is enabled, it discharges the bitline and causes the replica bitline output signal to go low.

4.5.3 Conclusion

In this section we looked at two schemes to generate the sense enable signal. The inverter chain and the replica bitline schemes were compared and it was found that the replica bitline scheme based on capacitance ratioing tends to track the bitline parasitics better across the process corners. Thus, this scheme was incorporated in our design.

4.6 Column Multiplexer

As shown in Chapter 3, the multiplexer resistance can affect the performance of the sense amplifiers. This is true mainly in the case of the current amplifier, where this resistance decreases the current signals, due to which the delay increases and the merit of using a current amplifier disappears. This is also true in the case of voltage amplifiers. Several publications [8, 43] propose increasing the width of the bitline multiplexer to improve delay. A few of the techniques that have been proposed have been discussed in chapter 3. However, all these are mainly directed towards current sensing and are area intensive.

Since we are using a voltage amplifier, these techniques are really not all that beneficial to us. We have thus tried to reduce the delay by increasing the width of the transistors. A disadvantage of increasing the width of the transistors is that it increases the parasitic capacitance. We also need to make sure that the area of the multiplexer does not exceed the pitch of the memory cell.

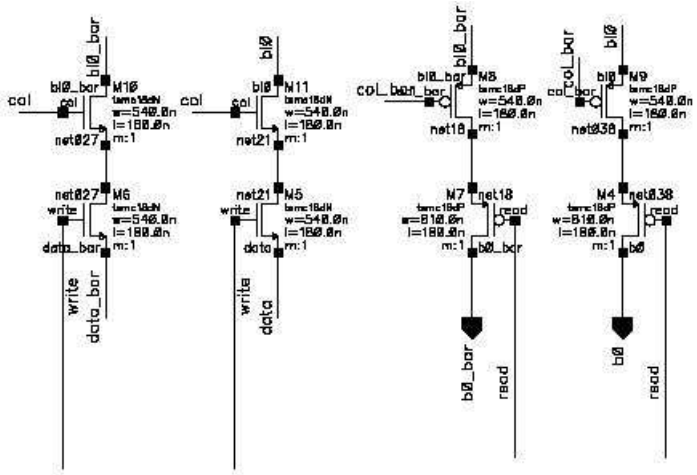


Figure 4.21: Schematic of column multiplexer

4.6.1 Design of the column multiplexer

The column multiplexer allows the data to either be written into the memory cell, or it passes the data from the bitlines onto the sense amplifier. The column multiplexer was designed using pass gates. The sizes of the transistors were chosen, so that they are wide enough to offer less resistance while being small enough to fit within the pitch of the memory cell.We have used pMOS transistors for the circuits that allow data to be read from the bitlines to the sense amplifiers, while we have used nMOS transistors to write the data onto the bitlines. We have to use one such set of column multiplexer's for each column of bitlines. Since we are reading/writing to 8 columns at a time we have created two blocks of 8 such column multiplexer's. Each block gets one input from the column decoder, which is then given to all the 8 multiplexer's in the block. The read and write signals along with the data line are also given as inputs to each block of the column multiplexer.Figure 4.21 shows the schematic of the column multiplexer for one bitline column.In the design of the memory cell we had seen that the pMOS transistor which connects the bitline to the sense amplifier should be sized as $810\eta m$. This is done so that the layout of the multiplexer fits within the pitch of the memory cell. nMOS transistors are used to

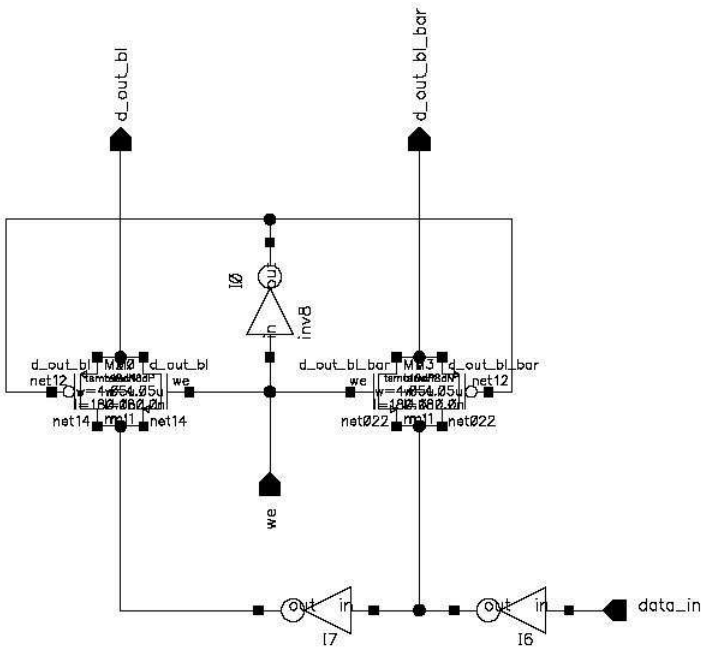


Figure 4.22: Schematic of Write Driver

write the data onto the bitlines. The working of the design is very simple. If the column decoder output is asserted and the write signal is enabled, then the data and $\overline{data}$ is driven on to the bitlines. If the column decoder output is asserted and the read signal is enabled, then the data on the bitlines is read out into the sense amplifier.SPICE simulations were used to verify this design.

4.7 Write Drivers

Figure 4.22 shows the schematic of the driver that is used to write data onto the bitlines. The capacitance of each column of bitline is in the range of 450fF. In chapter 3 we have shown a design which can be used to reduce the power dissipation due to writes. However, that scheme has quite a few drawbacks which have been explained in the chapter. The design also consumes quite a lot of area.

We have gone for a very simple design, where the data and the $\overline{data}$ are driven onto the bitlines mainly using inverters. Pass gate transistors which are enabled by the

write signal allow these signals to be put on the bitlines. The major constraint on the size of the inverters is the area of the memory cells. We need to ensure that the write drivers fit within its pitch width. Also there is no improvement in the performance of the circuit beyond a certain point and it only leads to self-loading. The most important design consideration is the size of the inverters which actually drive the data on to the bitlines.

4.8 Conclusion

In this chapter we have shown the design of a 4K SRAM macro. A simple monolithic architecture was implemented. The memory array is composed of 256 rows and 16 columns. The array is split into two blocks of 2K each, each block having 256 rows and 8 columns. Thus, a byte of data can be written and read from the array at any time. SCL based row decoders were implemented since they consume small area and power. A unique resetting technique for the word line pulses was implemented. This technique utilizes the replica bitline output signal and helps to minimize the word line pulse width to 0.7ns which in turn helps reduce the power dissipation. The 6 transistor model was used for the memory cells since it is more robust than the 4 transistor model. The alpha latch was chosen as the sense amplifier and was designed to that it could amplify signals as low as 50mV. The replica bitline scheme based on capacitance ratioing was used to enable the sense amplifier at the most optimum time (when the bitline voltage is 180mV). The schematics of all the designs along with their layouts have been given in Appendix A.

All the above designs were also tested and verified to make sure that they work as expected. The entire design was then simulated to verify its working. All the tests along with the results have been summarized in the next chapter.

Chapter 5

Testing and Verification

The designs implemented in the previous chapter were tested and the results verified. Each component of the design, such as the memory cell, the decoder and the sense amplifier was tested to make sure that the design was correct and that expected results were obtained. Each design was simulated across the four process corners. Finally, the entire 4K design was simulated and the results verified.

5.1 Memory Cell Testing

Figure 5.1 shows the setup that was used to test the working of the memory cell. A 256 row single column array of memory cells was used in the setup. Of these, only the first memory cell is used for the simulation purposes and the rest of the cells are used as dummy cells to emulate the capacitive load which has to be driven by the write drivers. Simple models were also used to determine the wire resistance and capacitance of the bitlines. The dummy cells are disabled by grounding their word line inputs. Along with the memory cells, even the precharge circuits, the bitline load circuit and the write driver circuits were tested.

Some of the critical areas that were tested included: 1) Making sure that the correct data gets written into the cells. This includes writing a 0 in a cell that contains a 1 and vice versa. 2) Making sure that the data is retained in the cells even after multiple reads. 3) The design of the bitline loads which pull up the bitlines after

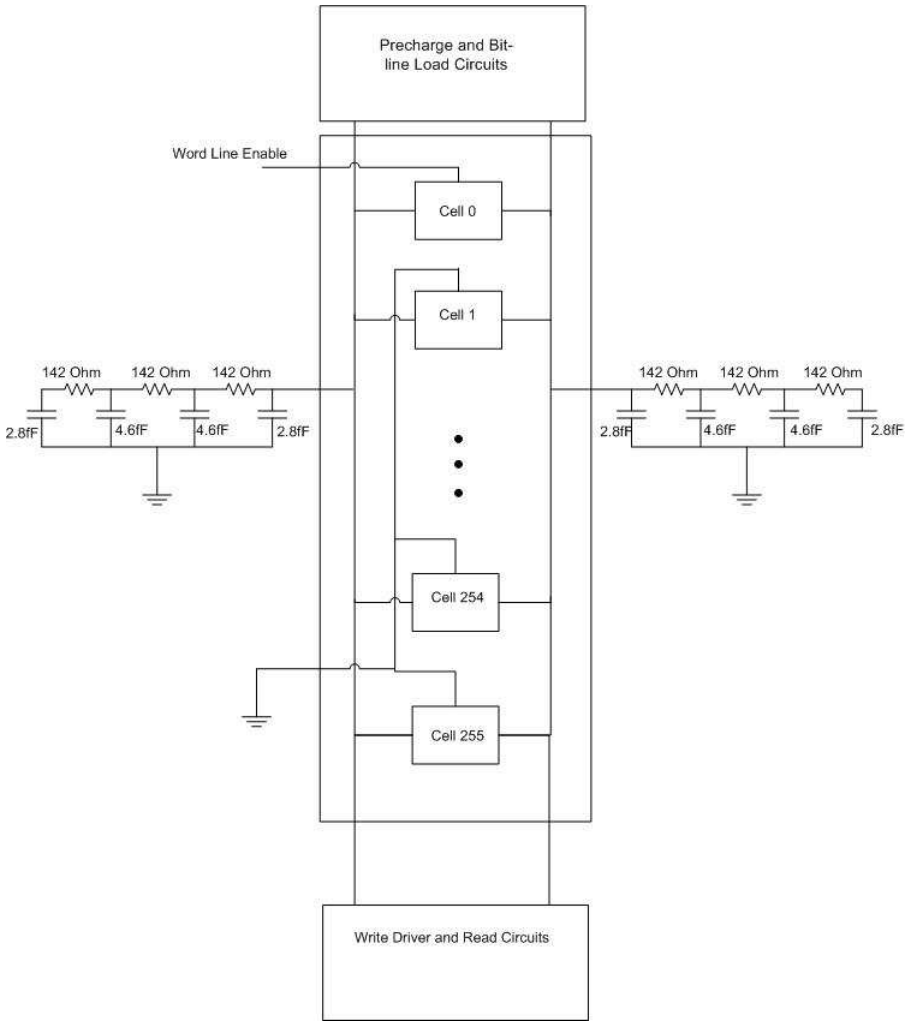


Figure 5.1: Setup for Memory Testing

a write operation is performed. The bitline load circuits have to be strong enough to pull up the bitlines fast enough for the next operation to be performed.
The bitline voltage was also varied to see the effect it has on the power dissipation. The lower bound for the bitline voltage was found to be 0.9V, below which the cell contents got destroyed. The design was then tested to make sure that it performs across the process corners.

5.1.1 RC interconnect delay of bitlines

When testing the memory cells, we also need to incorporate the effects due to interconnect resistance and capacitance. We have used simple models to incorporate this in our testing.
The bitlines have been laid out using Metal 1 layer in the $0.18\mu m$ technology. By dumping the tech file for this technology we were able to obtain the sheet resistance for the metal 1 layer. Once we know the sheet resistance we can calculate the resistance of the wire by the following formula.

$$Resistance = \square * L/W \tag{5.1}$$

$\square$ is the sheet resistance while W is the width of the metal layer and L is the length of the bitline. The sheet resistance was found to be 0.08Ω. The width of the metal layer is $0.27\mu m$ and the length of the bitlines was found to be $1.54\eta m$.

$$Resistance = 0.08 \times 1.54\eta/0.27\mu \tag{5.2}$$

$$Resistance = 426\Omega \tag{5.3}$$

The capacitance of the wire was calculated using the formula 5.4. We have spaced the bitlines by a distance greater than the minimum space required. Hence fringing capacitance is no longer an issue and is not calculated.

$$C_{wire} = \frac{\xi_{di}}{t_{di}} WL \tag{5.4}$$

Substituting the values we get the capacitance of the bitline wires to be

$$C_{wire} = 14.1fF \tag{5.5}$$

This gives us the lumped RC model of the wire. For simulating this in SPICE we approximated the distributed RC model by using the Π3 model. [16] has shown that the error of the Π3 model is less than 3%, which is sufficient for our applications.

5.1.2 Writing to a Memory Cell

For the writing operation, a sequence of 1's and 0's was written into the memory cell. This was done to make sure that the data is correctly written into the cell. The delay taken to write the data into the cell, as well as the bitline swing was recorded.

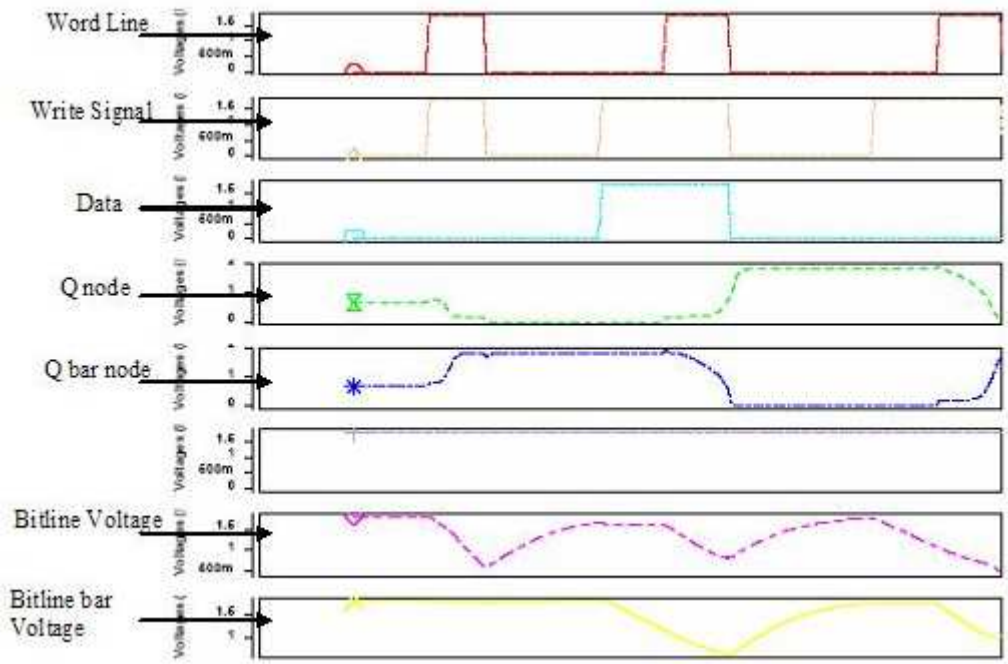


Figure 5.2: Writing Data into Memory Cell

The data was placed on the data line of the write driver circuit. The bitline loads were then disabled so that the data could be written on to the bitlines. The cell was selected by enabling the word line of the cell. The write signal was then enabled to write the data on to the bitlines. The word line was disabled after 0.7ns. This word line width was chosen, since this is the delay after which the decoders are reset by the replica bitline circuit in the 4K SRAM design. Initially a 0 was written in to the

cell, and after a 2ns delay a 1 was written into the same cell.The Figure 5.2 shows the waveforms which show the writing of the 1 on to a previously stored 0 followed by a 0 being written in to the same cell.These tests were then performed across the four process corners to make sure that the correct data is written into the cells. The

Process Corner	Delay from Word Line to Data Output	Maximum Bitline swing	
		Writing a 1 on a previous 0	Writing a 0 on a previous 1
Normal Conditions	1.05ns	0.183V	0.532V
Fast	0.256ns	1.3V	1.3V
Slow	1.15ns	0.107V	0.535V
FastN/SlowP	0.657ns	0.891V	1.01V
SlowN/FastP	0.649ns	0.891V	1.01V

Table 5.1: Results for simulations performed for Write operation

results are as shown in the Table 5.1. The delay measured is from the word line reaching 10% (0.18V) of its final value, to the cell voltage reaching 90% (1.62V) of its final value. These simulations were performed keeping the bitline voltage at 1.8V In a later section we will look at the effect of reducing this voltage on the delay and the power dissipation.

5.1.3 Reading from a Memory Cell

Another critical aspect of the memory cell design, is to make sure that read operations don't overwrite the contents of the memory cell. This can easily happen if the transistors are not properly sized. Assume that a 1 is stored at the Q node, while a 0 is stored at the $\overline{Q}$ node. During a read operation current will flow through the nMOS at the $\overline{Q}$ node. This in turn will increase the voltage at this node. If this voltage gets large enough, the nMOS at the Q node will start conducting and this can cause the memory cell contents to flip. The differential voltage between the bitlines should be around 180mV, so that it is correctly read by the sense amplifier.

The precharge circuits are enabled before a read operation so that they precharge the bitlines to 1.8V. The memory contents are then read by enabling the word line

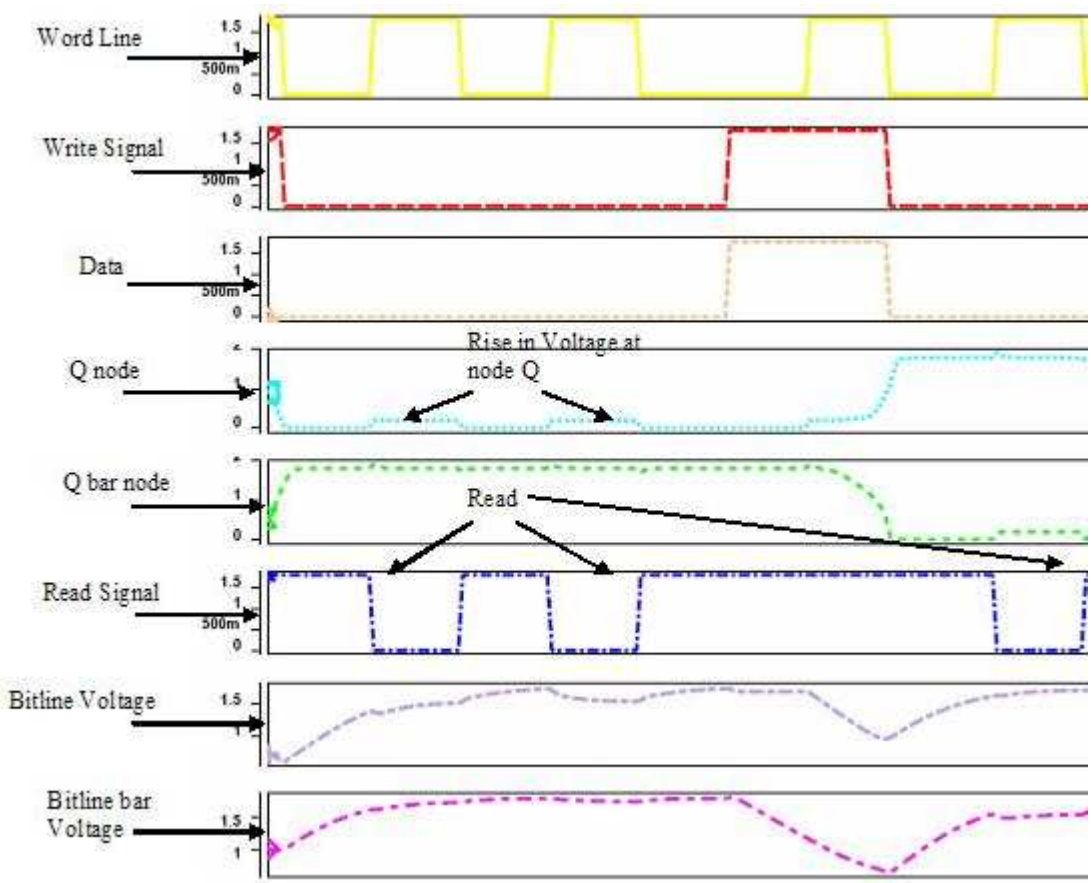


Figure 5.3: Reading Data from Memory Cell

of the cell, and by enabling the read signal. The word line pulse width is maintained at 0.7ns. This makes sure that the bitline voltage is around 180mV. The replica bitline scheme based on capacitance ratioing is used to enable the sense amplifier at the correct moment. 29 dummy cells were used in the replica bitline, so that when the sense amplifier was enabled, the differential voltage was around 180mV. For the purpose of testing, a 0 was written into the memory cell. This was followed by two consecutive reads. As seen in Figure 5.3, the read operation causes the voltage at node Q to go up slightly (around 150mV), but this is not enough to cause a change in the contents of the memory cell. The second read confirms that the original data was not lost.

The same set of tests were then performed at the process corners to make sure that the design works across them.

Process Corner	Bitline Differential Voltage
Normal Conditions	220mV
Fast	240mV
Slow	430mV
FastN/SlowP	310mV
SlowN/FastP	300mV

Table 5.2: Bitline Differential Voltages obtained during Read operations

The results obtained are as shown in Table 5.2. As seen the read operations performs across all the corners, with the maximum deviation from the optimal bitline voltage being 250mV.

Thus, from the above tests we can conclude that the memory cell design works correctly across all the process corners. Another interesting aspect to look at is how the reduction in the bitline voltage affects the operation.

5.1.4 Effects of varying bitline voltage

The bitline voltage can be varied to reduce the power dissipation. After any read or write operation the voltage to which the bitlines have to be precharged is reduced and this reduces the delay and the power dissipation. [8] have shown that there exists a lower limit to this voltage, below which the memory contents are lost. The analysis of this has already been discussed in Chapter3 and from the simulations run by [46] it was found that 0.68V was the lower limit for a 0.18μm technology. Table 5.3 shows the results that were obtained by varying the bitline voltages. The voltages were varied from 1.8V down to 0.9V. The power dissipation reduces from 4.6mW to 3.88mW. The reduction in delay, from the word line reaching 10% of its value to the output reaching 90% of its value, stabilizes after a certain point. Below 0.9V the data in the memory cells is lost. Figure 5.4 shows the memory cell contents when the bitline voltage is at 900mV and at 800mV. As seen, when the bitline voltage is

Bitline Voltage	Delay from Word Line to Data Output	Maximum bitline swing		Bitline voltage (Read)	Power Dissipation
		Writing 1 on a previous 0	Writing 0 on a previous 1		
1.8V	1.05ns	0.783V	0.932V	0.24V	4.6mW
1.5V	0.946ns	0.618V	0.906V	0.34V	4.17mW
1.3V	0.840ns	0.775V	0.890V	0.61V	4.02mW
1.1V	0.966ns	0.756V	0.796V	0.75V	4.19mW
1V	0.96ns	0.683V	0.361	0.57V	4.15mW
0.9V	0.968ns	0.681V	0.360V	0.55V	3.88mW

Table 5.3: Simulations showing effects of varying Bitline Voltages

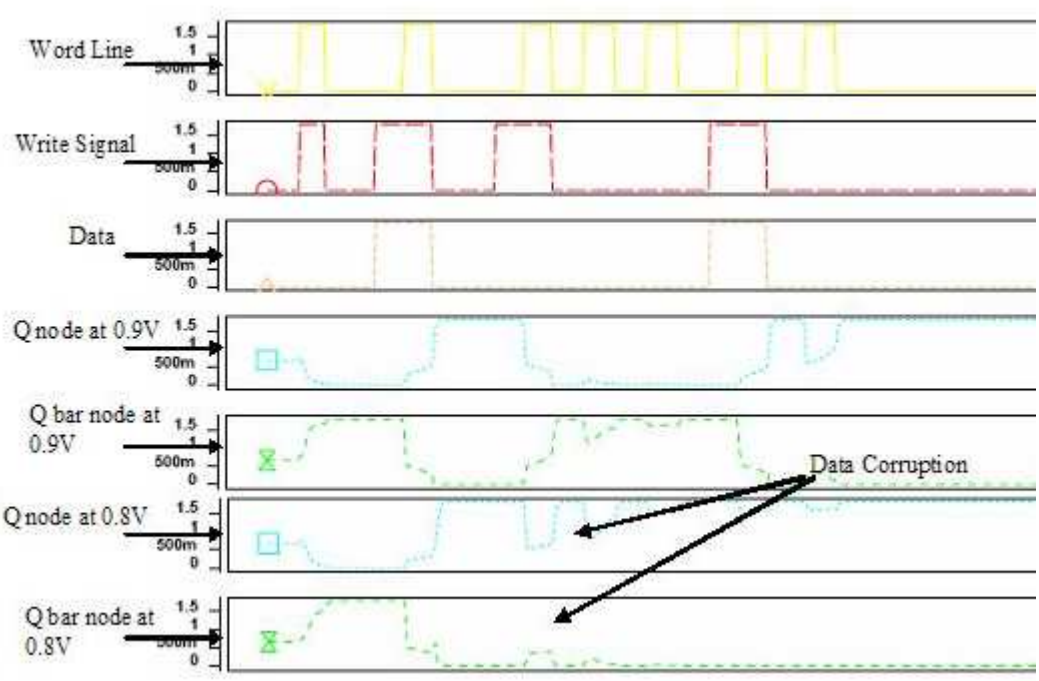


Figure 5.4: Memory Cell contents when Bitline voltage is 900mV and 800mV

reduced to 800mV, the contents of the memory cell are lost once the write signal is removed. Thus, for our design we can lower the bitline voltage down to a minimum of 900mV.

5.1.5 Conclusion

We can conclude from the tests that the memory cell design works as expected across all the process variations. The write driver is able to drive the data on to the bitlines, and into the memory cells. Also, the bitline load circuits are able to pull up the bitlines fast enough for the next operation to be performed. We have also confirmed that reading data from the cells leaves the original data intact, and that the precharge circuits are able to precharge the bitlines before the read operation is performed.

We can reduce the power dissipation by reducing the bitline voltage down to a minimum of 900mV.

5.2 Row Decoder Testing

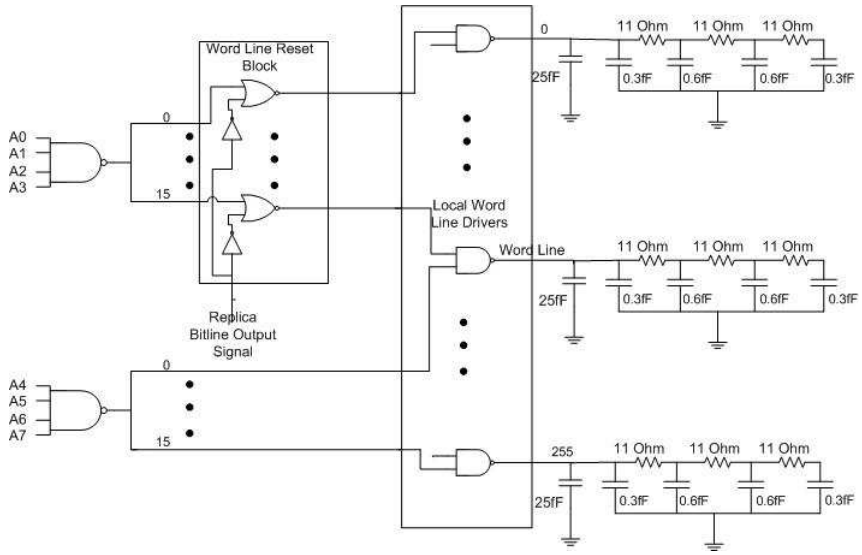


Figure 5.5: Setup to test working of Row Decoders

Figure 5.5 shows the setup that was used to test the working of the row decoders. The decoder structure is similar to the one used in our design. It consists of two sets of 4-to-16 NOR type decoders based on the designs described in [12]. One set of the

outputs goes through the Word Line reset circuits which also have the replica bitline output signal as the other set of inputs. The two sets of 16 outputs are then combined by the Local Word Line drivers to obtain 256 row outputs. The simulations were run across the process corners.

The replica bitline reset signal was applied so that the word line pulse width is around 0.7ns. So, initially a particular address was put on the address lines. The replica bitline reset signal was then asserted after 0.7ns so that the pulse width was close 0.7ns. This is the word line pulse width that is used in our final design.

5.2.1 RC interconnect delay of word lines

When testing the decoder, we also need to incorporate the effects of the word line interconnect resistance and capacitance. We have used simple models described in Section 5.1.1

The word lines have been laid out using Metal 2 layer in the $0.18\mu m$ technology. By dumping the tech file for this technology we were able to obtain the sheet resistance for the metal 2 layer. The equation 5.1 was used to find the resistance of the metal2 word lines. The sheet resistance was found to be 0.08Ω. The width used for the word lines is $0.36\mu m$ and the length of the word line was found to be $144\mu m$. This is the length of the word line for the entire memory array, i.e. 16 columns. In our layout we have placed the decoders in the center, i.e. between the two sets of 8-bit columns.

$$Resistance = 0.08 \times 144\mu/0.36\mu \tag{5.6}$$

$$Resistance = 32\Omega \tag{5.7}$$

The capacitance of the wire was calculated using the equation 5.4. Substituting the values we get the capacitance of the bitline wires to be

$$C_{wordline} = 1.78fF \tag{5.8}$$

This gives us the lumped RC model of the wire. For simulating this in SPICE we approximated the distributed RC model by using the $\Pi 3$ model. [16] has shown that the error of the $\Pi 3$ model is less than 3%, which is sufficient for our applications.

5.2.2 Results

Process Corner/Temperature	Delay from Input to Output	Power Dissipation
Normal/25C	0.62ns	1.44mW
Normal/100C	0.78ns	1.47mW
Fast/25C	0.47ns	1.07mW
Fast/100C	0.52ns	1.10mW
Slow/25C	0.70ns	1.10mW
Slow/100C	0.79ns	1.13mW
FastN/SlowP/25C	0.56ns	1.10mW
FastN/SlowP/100C	0.63ns	1.11mW
SlowN/FastP/25C	0.48ns	1.06mW
SlowN/FastP/100C	0.63ns	1.11mW

Table 5.4: Simulation Results for Row Decoder Testing

Table 5.4 shows the results that were obtained from the simulations. The delay represents the delay between the word line output to reach 90% of its final value and the address inputs reaching 10% of their final value. The simulations were run at both 25C as well as 100C, and as seen the designs work at that temperature as well. The average delay time is around 0.6ns.

5.3 Sense Amplifier Testing

The working of the sense amplifier was simulated to make sure that it could amplify the differential voltages which it reads on the bitlines. The critical aspect in the testing was to make sure that the amplifier design could amplify voltages which are smaller than 180mV.

5.3.1 Test Setup

The sense amplifier is enabled by using the replica bitline scheme based on capacitance ratioing as described in Chapter 4. We have designed the replica bitline so that it enables the sense amplifier when the bitline voltage is 180mV. However we have seen in the previous chapter how process variations can affect this design. The sense

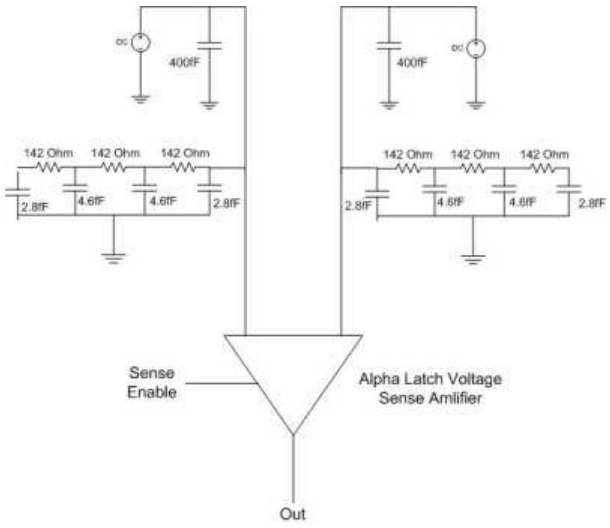


Figure 5.6: Test setup for sense amplifier

amplifier was tested by varying the bitline voltage. Figure 5.6 shows the test setup used. Two voltage sources were used to vary the bitline voltage. The capacitance of the memory cells for a column was extracted from the layout and was found to be around 400fF. The parasitics of the bitlines have already been calculated in section 5.1 and these were used for this test as well.

The simulations were run by varying the differential bitline voltages from 50mV to 200mV. The sense enable signal was used to enable the sense amplifier at the appropriate time. To make sure that the design also works across all process corners, we ran the simulations by keeping the differential bitline voltage at 150mV and running the simulations over the process corners.

5.3.2 Results

Table 5.5 shows the simulation results obtained by varying the bitline voltages. The bitline voltage was varied from 1.5V to 1.7V while the $\overline{bitline}$ voltage was maintained at 1.75V. We thus varied the differential bitline voltage from 250mV to 50mV. The average delay time, from when the sense enable signal is asserted to when the output reaches 90% of its value, reduces with an increase in the differential bitline

Bitline Voltage	Delay between assertion of read signal and output
200mV	0.432ns
180mV	0.440ns
150mV	0.452ns
120mV	0.459ns
90mV	0.471ns
50mV	0.485ns

Table 5.5: Effects of varying bitline voltage on delay of sense amplifier

voltage. However, the main purpose of these simulations was to make sure that the sense amplifier is able to perform for bitline voltages which are less than 180mV. Table 5.6 shows the variation caused due to the different process variations. The bit-

Process Corner	Delay between assertion of read signal and output
Normal	0.452ns
Fast	0.412ns
Slow	0.475ns
FastN/SlowP	0.431ns
SlowN/FastP	0.439ns

Table 5.6: Effects of process variations on delay of sense amplifier

line voltage was maintained at 150mV, and the simulations were run over the process corners. The average delay of the sense amplifier is around 4.4ns.

5.3.3 Conclusion

The simulations were run to make sure that the sense amplifier is able to perform across the process corners, and also with a varying bitline voltage. From the results it is evident that the sense amplifier design meets these criterion. The amplifier is able to amplify voltages which were as small as 50mV.

From the above sections it is evident that the three main designs - The memory cell,

the row decoder and the sense amplifier, work as expected. In the next section we shall shows the results obtained by testing the entire design as a whole.

5.4 Testing of the 4K design

The design that has been described in Chapter 4 was tested by writing different combinations of data into the memory array and then reading it out to make sure that the design worked. The models for the RC delay of the word lines and the bitlines have already been calculated in the previous sections, and these models were incorporated into the schematic.

5.4.1 Test setup

A clock speed of 2ns was used to run the simulations of the entire design. A byte of data is written into the memory array and then this data is read out to make sure that the data was correctly stored. Some of the main criteria while testing were: 1) Making sure that the correct word line is asserted by the row decoders. 2) The correct data gets written into the memory cells. An important aspect of this is to make sure that the write drivers are able to drive the bitlines and store the data in the memory cells. The bitline load circuits should be able to pull the bitlines up to V_{DD} once the write signal is de-asserted. 3) New data should be able to overwrite a previously stored data. 4) When reading the memory array we need to make sure that the stored data does not get corrupted. 5) Various timing considerations also need to be taken into account. The replica bitline output should enable the sense amplifier when the bitline voltage is close to 180mV, and the word line should then be de-asserted. The results obtained after running these tests are described in the next few sections.

5.4.2 Writing data into the memory array

Different sequences of 1's and 0's were written into the memory array to make sure that the write circuitry works. Figure 5.7 shows the waveforms that were obtained

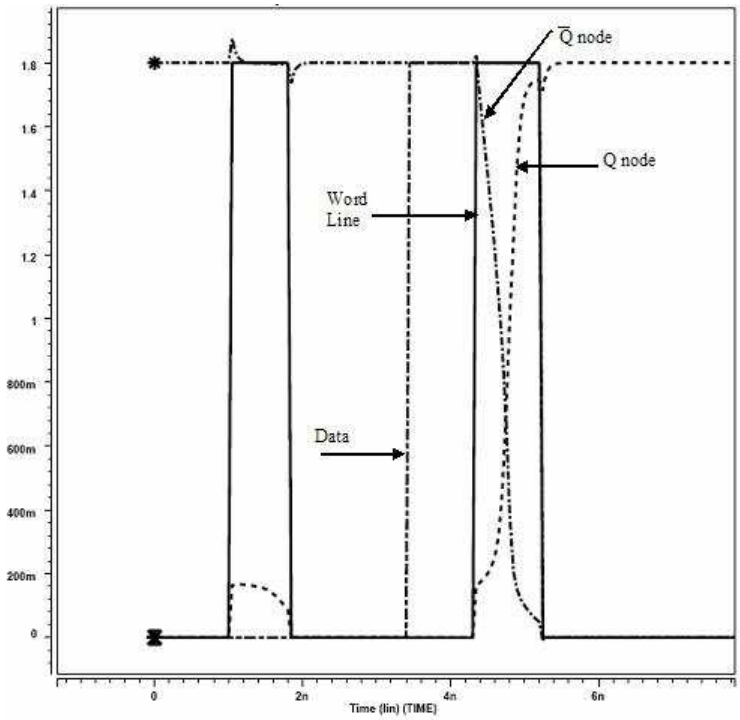


Figure 5.7: Writing a 1 into the memory cell

when writing a 1 into the memory cell. As seen the, once the write signal is asserted and the word line is pulled high, the Q node starts to rise to V_{DD} while the $\overline{Q}$ node starts dropping down to Gnd. The write access time is the time elapsed between a write request and the final writing of the input data into the memory. From simulations this was found to be 0.62ns. The write cycle is the minimum time required between write cycles. From the simulations this was found to be 1.2ns. The primary reason why the write cycle is greater than the write access time is because once the write signal is de-asserted the bitlines have to be pulled back to V_{DD} by the bitline load circuits. The bitlines have large capacitances and this adds to the delay.

5.4.3 Reading data from the memory array

The data which is present in the memory array needs to be read out by the sense amplifier circuits which have to be enabled by the replica bitline circuit. During the

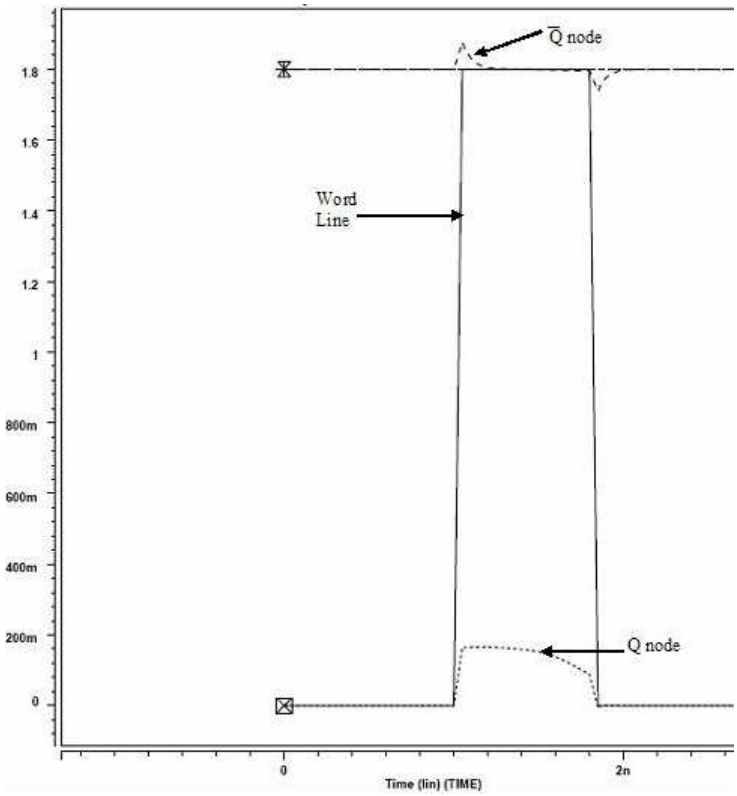


Figure 5.8: Reading data from the memory cell

read operation there are chances of the data getting destroyed. Figure 5.8 shows the waveforms that were obtained from the simulations.As seen when reading a 0 from the memory array, the voltage at node Q rises up to 180mV. This is however not sufficient to flip the contents of the memory cell. The read access time is the time taken to retrieve data from the memory cell. For our design the read access time was found to be 0.83ns. The read cycle is the minimum time required between successive reads. For our design this was found to be 1.4ns. The read cycle time is greater than the read access time since the bitlines need to be precharged before a read operation can be performed.

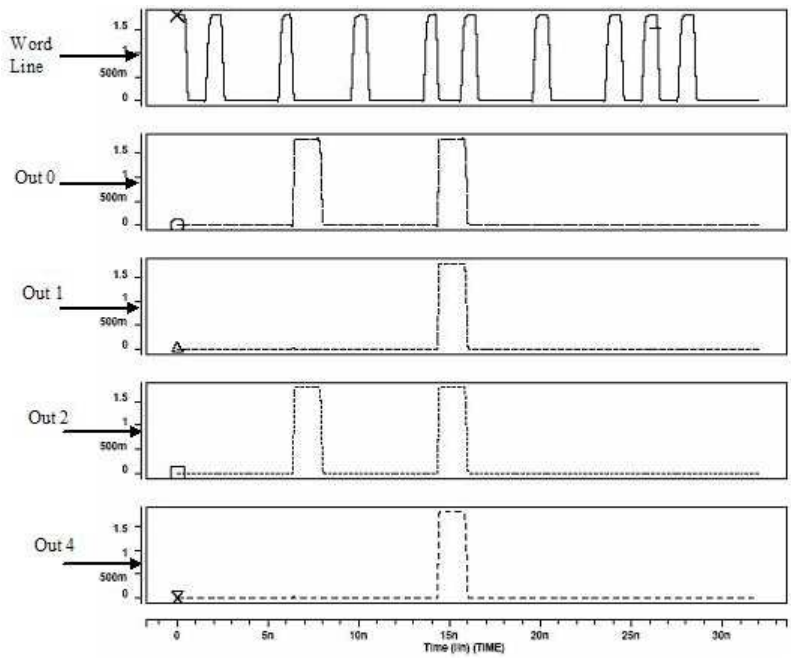


Figure 5.9: Simulation results

5.4.4 Results

Figure 5.9 shows the output waveform from one set of simulations that were run. The waveforms show the snapshot obtained for 4 outputs. Initially a byte of data consisting of a sequence of alternate 1's and 0's was written into the memory array. This data was then overwritten by a sequence of all 1's. The design was run at 500Mhz and at 1.8V V_{DD}. The results obtained from the simulations have been summarized in Table 5.7.

5.5 Conclusion

The design described in chapter 4 was tested to make sure that the designs worked as expected. Each block of the 4K SRAM was independently tested and the results verified. The designs were simulated across the process corners to make sure that they can withstand the process variations. Finally the entire design as a whole was

Clock Speed	500Mhz
Power Dissipation	26.3mW
Read-access time	0.83ns
Read-cycle time	1.4ns
Write-access time	0.62ns
Write-cycle time	1.2ns

Table 5.7: Final results obtained from simulations

simulated. For each design the RC delay of the interconnect wires was calculated using simple equations and these values were incorporated in the simulations.

Chapter 6

Conclusion

In this thesis we have looked at the different tradeoffs involved in the design of an SRAM. We have looked at the three major components of the SRAM: The memory cell, the row decoders and the sense amplifiers. The tradeoffs involved in the design of each component was studied and explored.

In Chapter 2 we looked at an overview of an SRAM and its components. In Chapter 3 we went into the details of each component. The different circuit techniques to reduce the logical effort and the power dissipation for the row decoders were studied. By running simulations we concluded that pulsed decoders are definitely a lot faster than conventional decoders though this does come at the cost of power dissipation. Various circuit techniques such as the SCL based decoders and the NOR based decoders were also described.
The sense amplifiers can be split into two main categories: The voltage sense amplifiers and the current sense amplifiers. In Chapter 3 we looked at the basic working principles of each category and some circuit styles to implement them.

In Chapter 4 we have described the 4K design that was implemented by us. A novel resetting scheme was looked at, which helps to reset the decoders at the most optimum time. This reduces the power dissipation of the design. This scheme uses the output from the replica bitline circuit which is used to enable the sense amplifier. Thus, the increase in area by using this scheme is very small. The power savings occur since this output resets the decoder at the optimum time, thus reducing the

differential bitline voltage to a minimum. In Chapter 4 we have described the design and the transistor sizing for each block.
The row decoders consist of 3 parts which include the SCL based predecoders, the word line reset block and the local word line drivers. For the memory cells, we have used the 6T based cells as they have a much better SNM compared to the 4T cells. The 6T cells have been designed keeping the memory cell density in mind and are made by keeping a cell ratio of 2. This gives us a SNM of around 600mV. A voltage sense amplifier (Alpha Latch) is used for the sensing purposes. The sense amplifier is designed keeping the pitch of the memory cell in mind, since we want the amplifier to fit within the area of the memory array. The replica bitline scheme based on capacitance ratioing is used to enable the sense amplifier. This same output is used to reset the row decoders as well.

In Chapter 5 we have looked at the results obtained by testing the designs described in Chapter 4. We ran extensive tests on all the designs to make sure that they performed as expected across the process corners. Finally the entire 4K design was tested at 500Mhz.

Some of the future work involves considering the effects of noise on the design. We have designed the 6T cells with a SNM of 600mV, but we need to simulate the design and see how noise affects the memory contents. Also, we need to see how noise can affect the output of the sense amplifiers. To see the advantages of using different architectures such as the DWL and the HWD architectures, we need to build a bigger memory array. We can also see the tradeoffs involved in using a current sense amplifier especially for larger designs.

Appendix A

Schematics

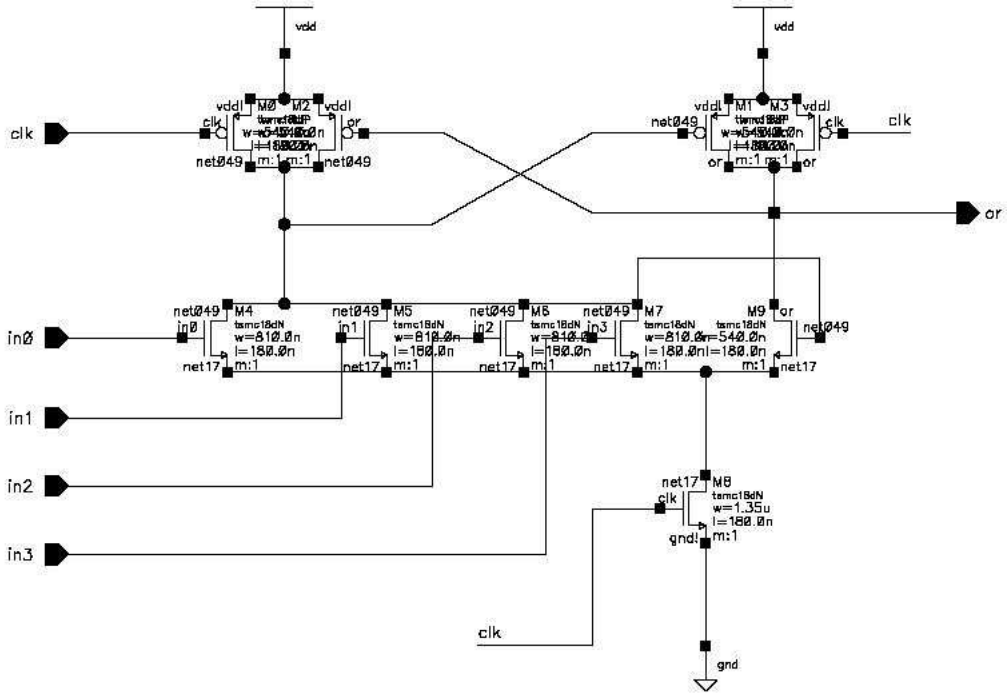


Figure A.1: Schematic of SCL NOR based pre-decoder

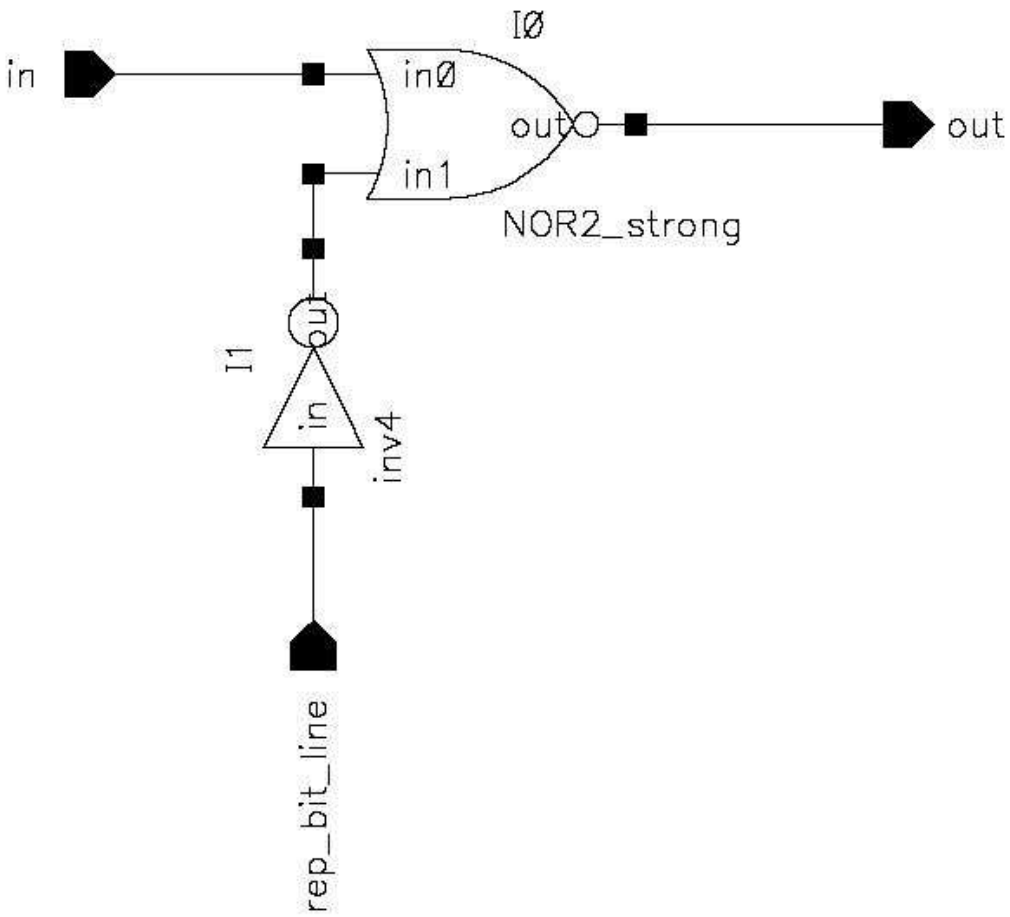


Figure A.2: Word Line Reset Circuit: Zoom 1

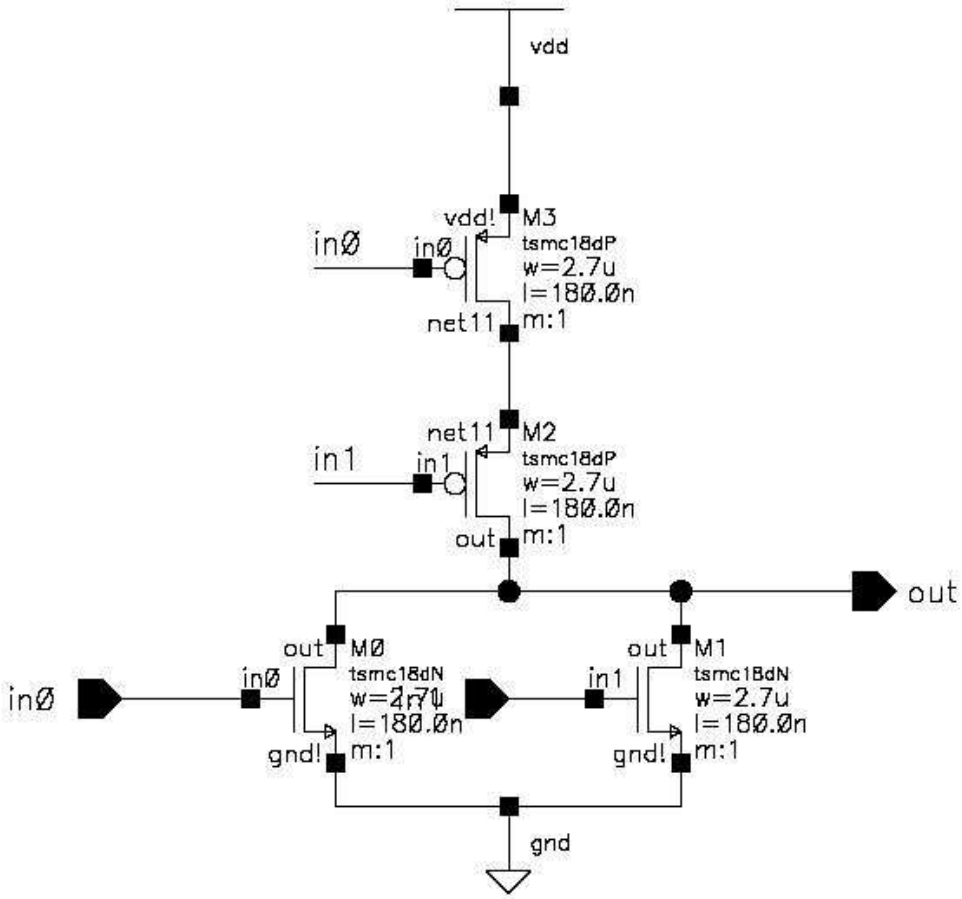


Figure A.3: Word Line Reset Circuit: Nor gate

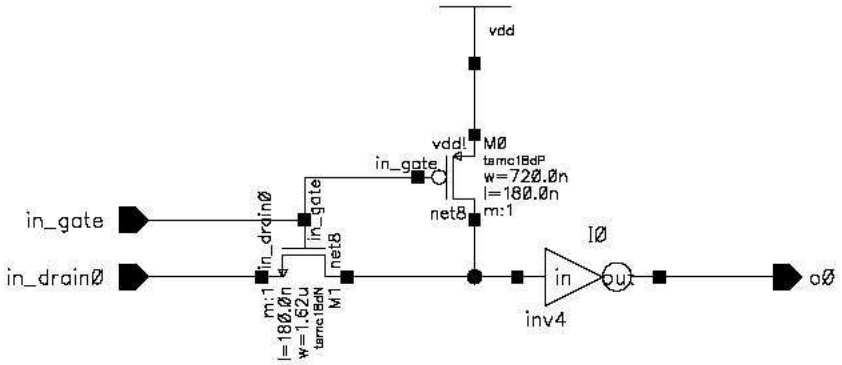


Figure A.4: Local Word Line Driver

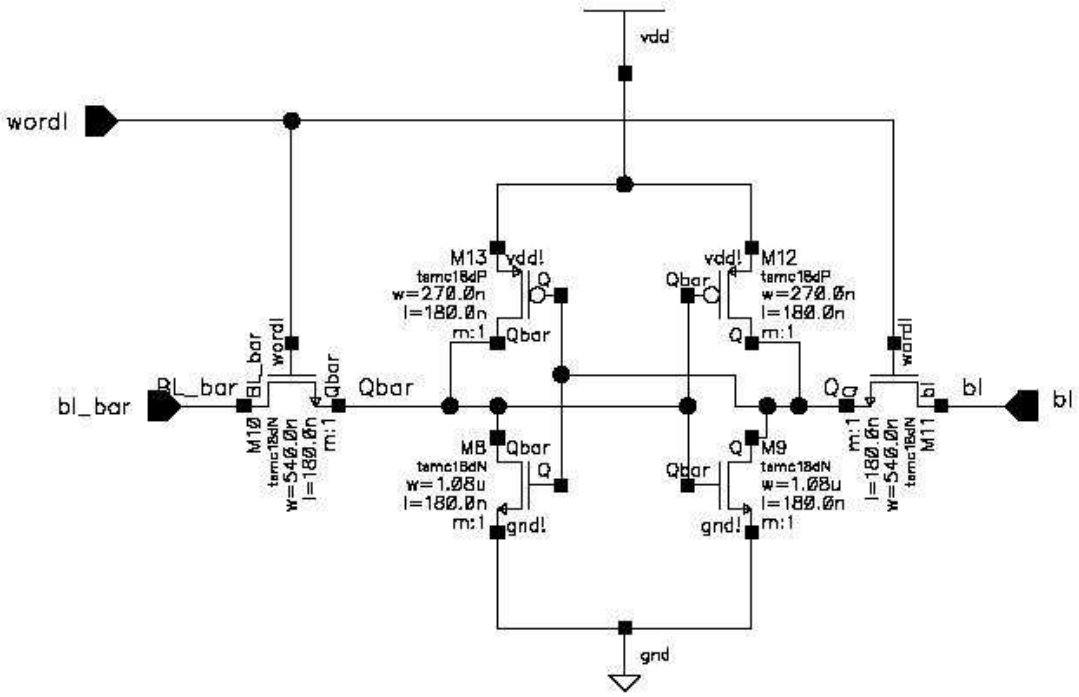


Figure A.5: Six Transistor Memory Cell

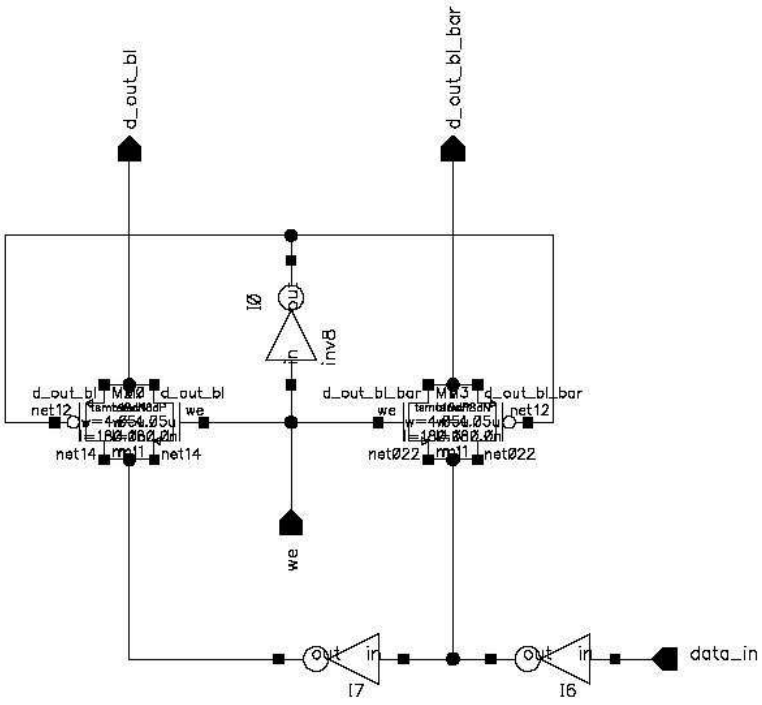


Figure A.6: Write Driver Circuit

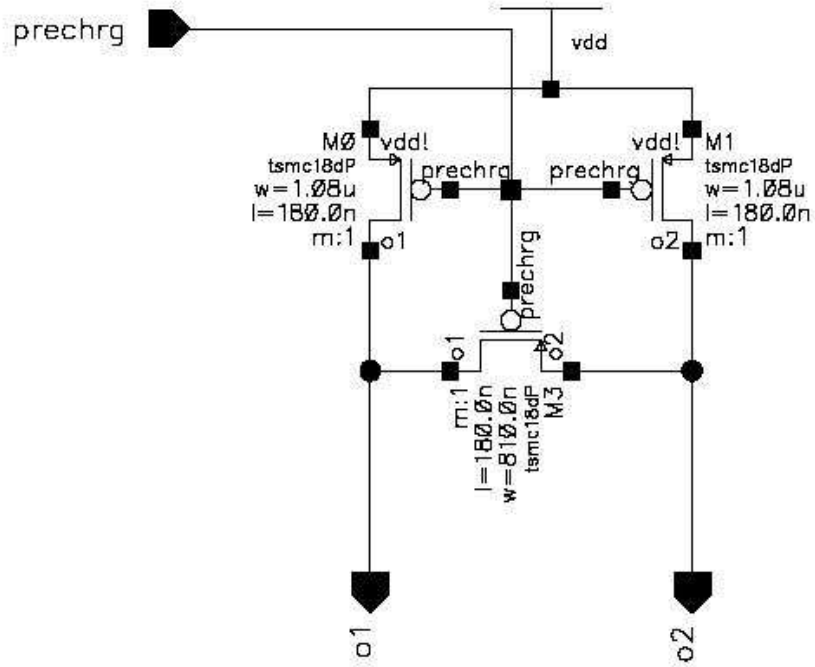


Figure A.7: Precharge Circuit for Bitlines

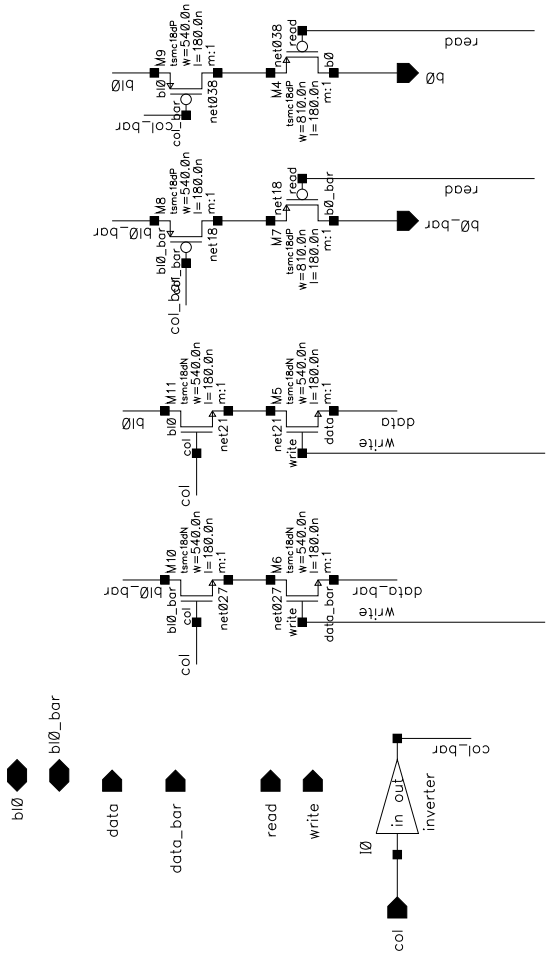


Figure A.8: Column Multiplexers

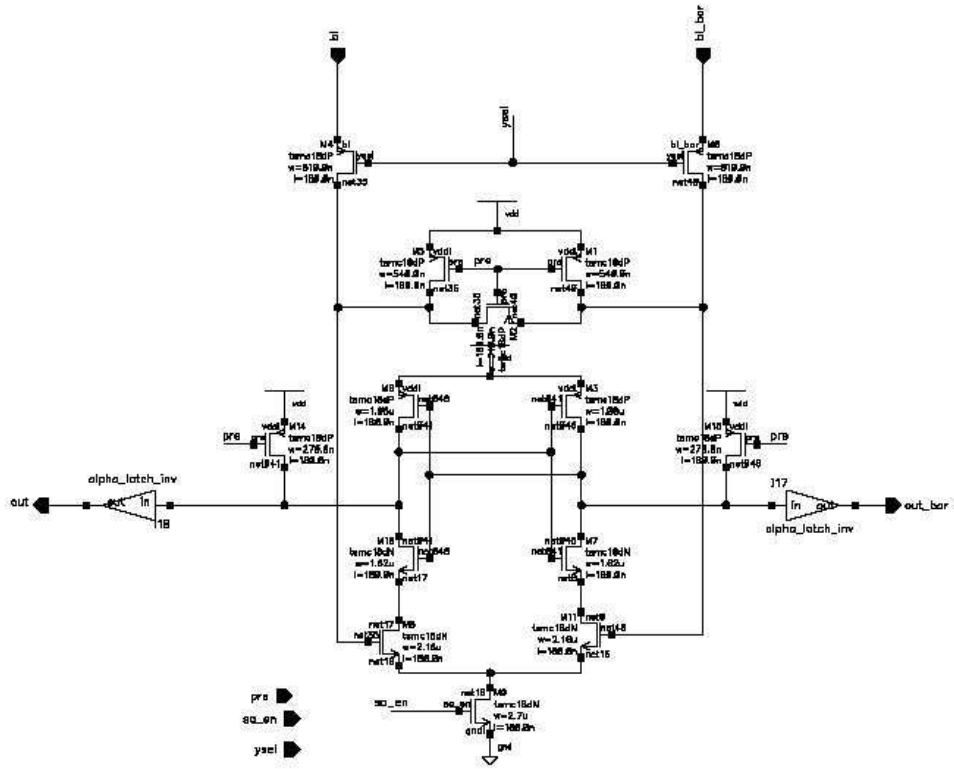


Figure A.9: Voltage Sense Amplifier

Appendix B

Layout

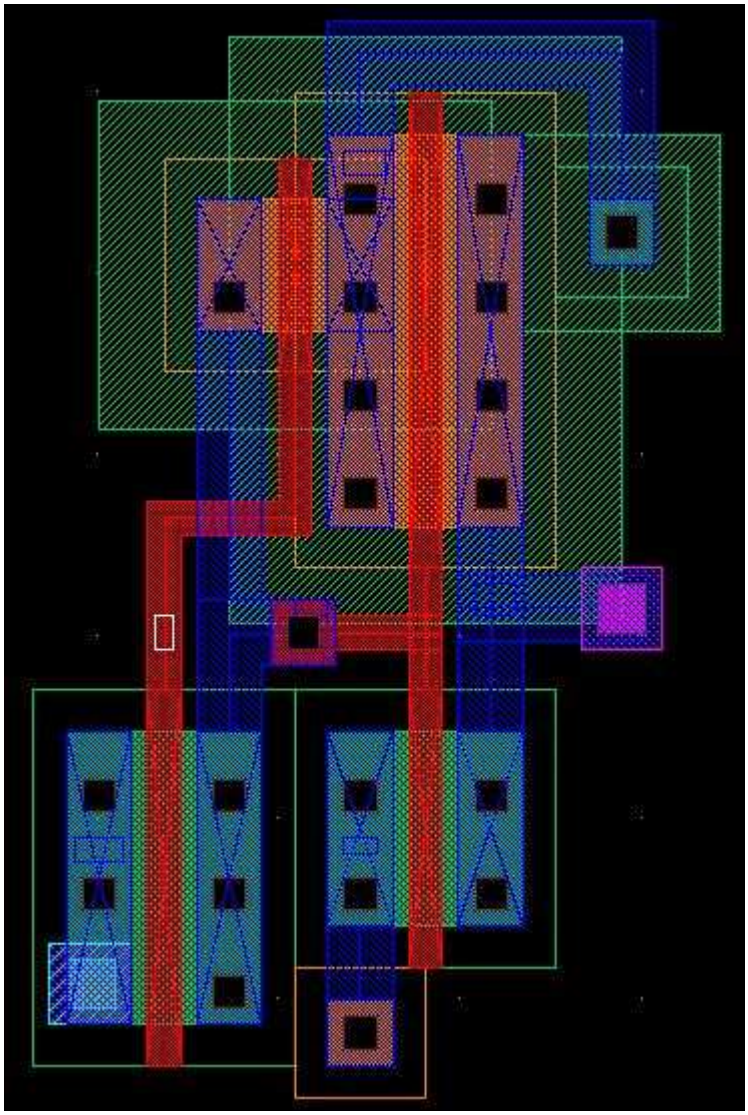


Figure B.1: Layout of SCL NOR based pre-decoder

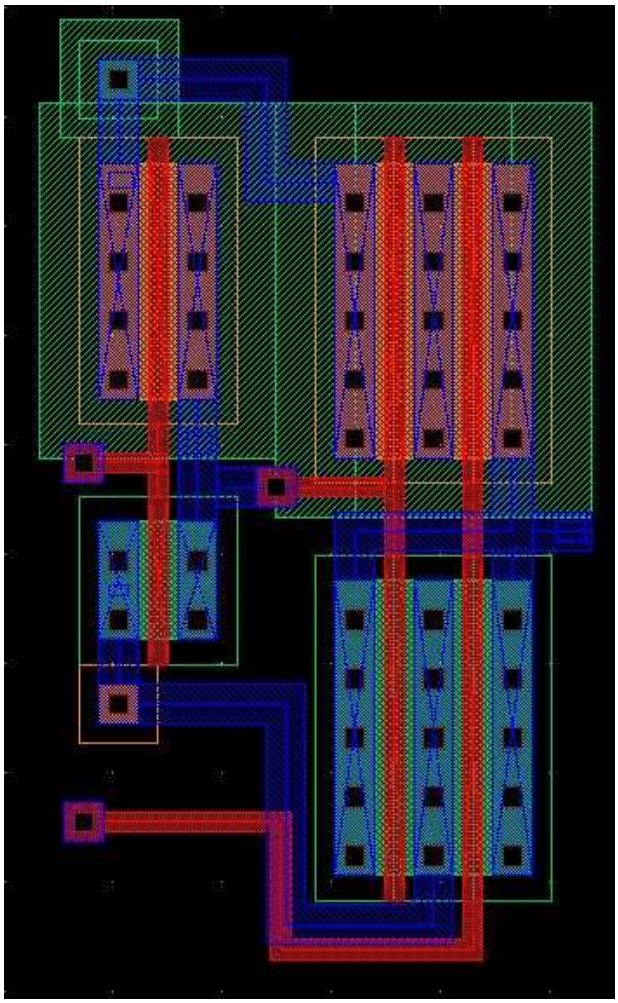


Figure B.2: Layout of NOR gate used in Word Line Reset Block

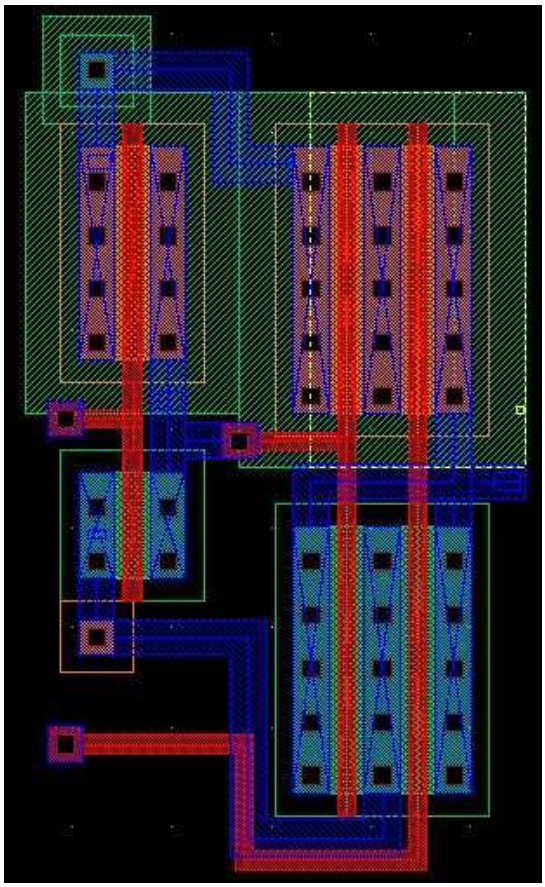


Figure B.3: Layout of Local Word Line Driver

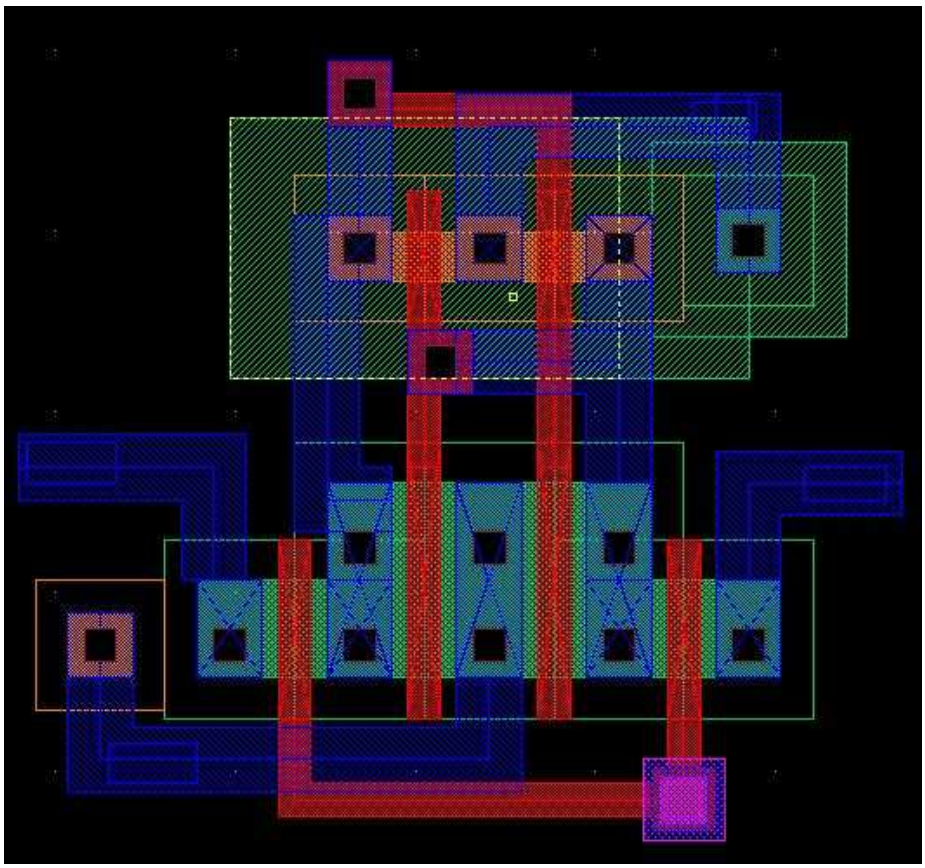


Figure B.4: Layout of 6T memory cell

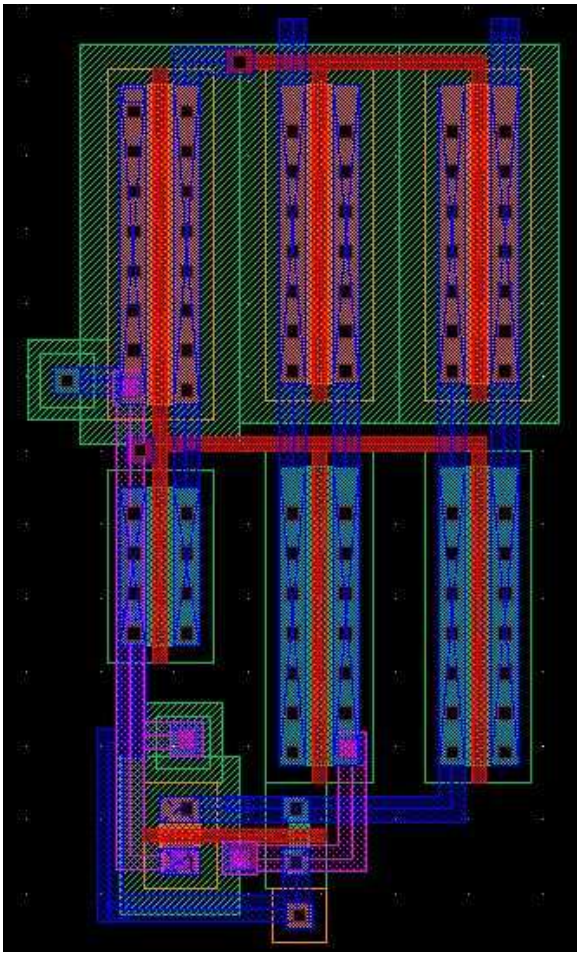


Figure B.5: Layout of Write Driver

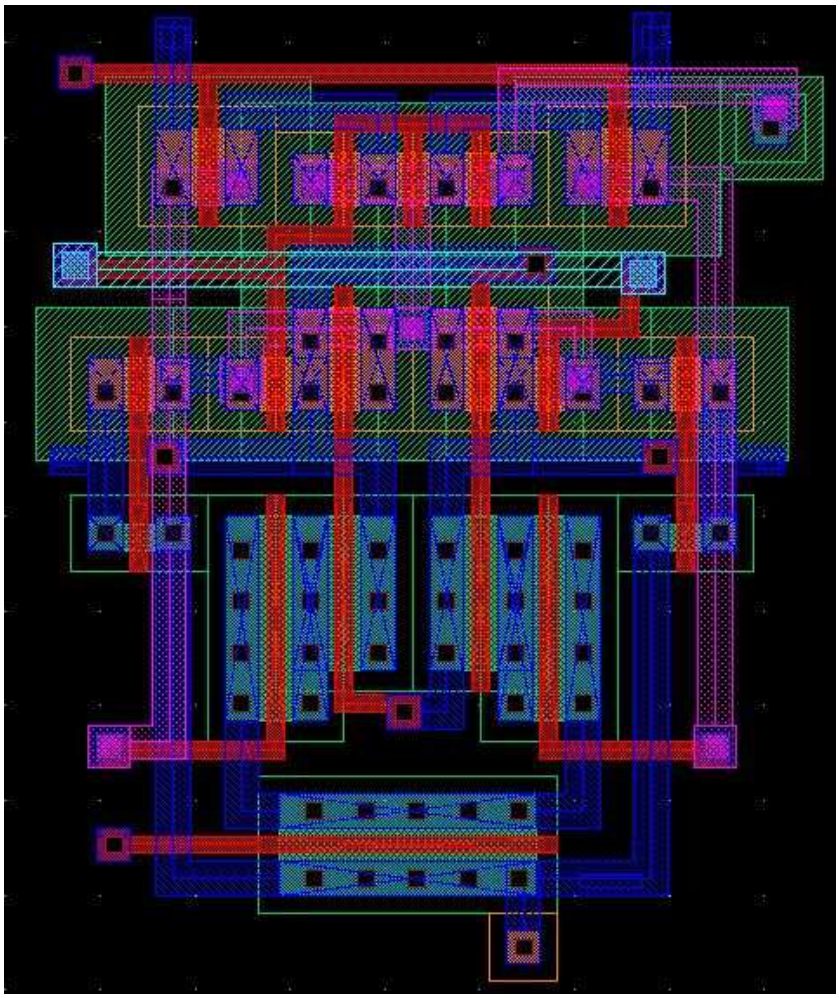


Figure B.6: Layout of Voltage Sense Amplifier

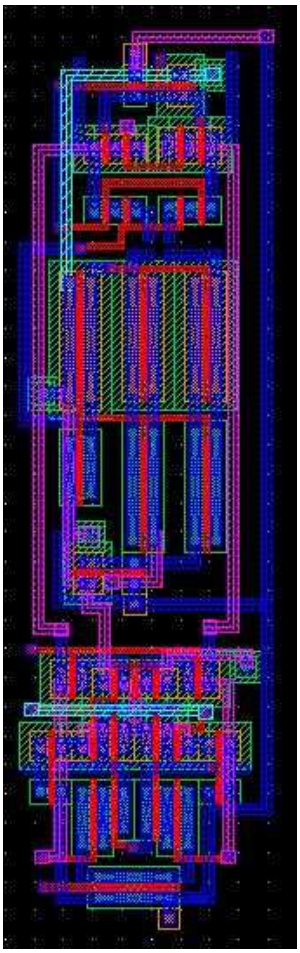


Figure B.7: Layout of Column Multiplexer, Write Driver and Sense Amplifier

Bibliography

[1] B. Amrutur and M. Horowitz. Techniques to reduce power in fast wide memories. *IEEE Symposium on Low Power Electronics, Digest of Technical Papers*, 1994.

[2] B.S.Amrutur and M.A. Horowitz. A replica technique for word line and sense control in low-power srams. *IEEE Journal of Solid State Circuits*, 33(8):1208–1219, 1998.

[3] B.S.Amrutur and M.A. Horowitz. Fast low-power decoders for rams. *IEEE Journal of Solid State Circuits*, 36(10):1506–1515, 2001.

[4] J.B.Kuo B.Wang. A novel two-port 6t cmos sram cell structure for low-voltage vlsi sram with single-bit-line simultaneous read-and-write access (sblsrwa)capability. *The 2000 IEEE International Symposium on Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva.*, 5, 2000.

[5] B. S. Cherkauer and E. G. Friedman. A unified design methodology for cmos tapered buffers. *IEEE Journal of Solid State Circuits*, 3(1):99–110, March 1995.

[6] A.P. Chandrakasan et. al. Low-power cmos digital design. *IEEE Journal of Solid State Circuits*, 27(4):473–484, 1992.

[7] B. Gieseke et. al. A 600 mhz superscalar risc microprocessor with out-of-order execution. *ISSC Dig. Tech. Papers*, Feb. 1997.

[8] Bernhard Wicht et. al. Analysis and compensation of the bitline multiplexer in sram current sense amplifiers. *IEEE Journal of Solid-State Circuits*, 36(11):1745–1755, Nov. 2001.

[9] Craig Lage et. al. Advanced sram technology - the race between 4t and 6t cells. *International Electron Devices Meeting*, 1996.

[10] G. Braceras et. al. A 350mhz 3.3v 4mb sram fabricated in a 0.3mm cmos process. *IEEE International Solid State Circuits Conference, Digest of Technical Papers*, 1997.

[11] G.G. Shahidi et. al. Partially-depleted soi technology for digital logic. *ISSCC Digest of Technical Papers*, 1996.

[12] H. Nambu et. al. A 1.8ns access, 550mhz 4.5mb cmos sram. *IEEE International Solid State Circuits Conference, Digest of Technical Papers*, 1998.

[13] H. Park et. al. A 833mb/s 2.5v 4mb double data rate sram. *IEEE International Solid State Circuits Conference, Digest of Technical Papers*, 1998.

[14] Hirotoshi Sato et. al. A 500-mhz pipelined burst sram with improved ser immunity. *IEEE Journal of Solid State Circuits*, 34(11):1571–1578, 1999.

[15] J. Choi et. al. Design of cmos tapered buffer for minimum power-delay product. *IEEE Journal of Solid State Circuits*, 29(9):1142–1145, September 1994.

[16] J. Rabaey et. al. Digital integrated circuits - a design perspective. 2003.

[17] K. J. Schultz et. al. Low-supply-noise low-power embedded modular sram for mixed analog-digital ics. *Proceedings of 1992 IEEE Custom Integrated Circuits Conference*, 1992.

[18] K. Nakamura et. al. A 500mhz 4mb cmos pipeline-burst cache sram with point-to-point noise reduction coding i/o. *IEEE International Solid State Circuits Conference, Digest of Technical Papers*, 1997.

[19] K. Osada et. al. A 2ns access, 285mhz, two-port cache macro using double global bit-line pairs. *ISSCC Digest of Technical Papers*, 1997.

[20] K Sasaki et. al. A 7-ns 140mw 1mb cmos sram with current sense amplifier. *IEEE Journal of Solid-State Circuits*, 27(11):1511–1518, Nov. 1992.

[21] Kevin Zhang et. al. A 15-ns 16mb cmos sram with interdigitated bit-line architecture. *IEEE Journal of Solid State Circuits*, 27(11):1497–1503, 2000.

[22] K.W. Mai et al. Low-power sram design using half-swing pulse-mode techniques. *ISSCC Digest of Technical Papers*, 33(11):1659–1671, 1998.

[23] M Ishida et. al. A novel 6t-sram cell technology designed with rectangular patterns scalable beyond 0.18 μm generation and desirable for ultra high speed operation. *International Electron Devices Meeting*, 1998.

[24] M. Matsumiya et. al. A 15-ns 16mb cmos sram with interdigited bit-line architecture. *IEEE Journal of Solid-State Circuits*, 27(11):1497–1503, Nov. 1992.

[25] M. Yoshimoto et. al. A 64kb cmos ram with divided word line structure. *IEEE International Solid State Circuits Conference, Digest of Technical Papers*, 1983.

[26] O. Minato et. al. A 20ns 64k cmos sram. *IEEE International Solid State Circuits Conference, Digest of Technical Papers*, 1984.

[27] S. Subbanna et. al. A high-density 6.9 sq. μm embedded sram cell in a high-performance 0.25μm-generation cmos logic technology. *IEDM Technical Digest*, 1996.

[28] S. Yamamoto et. al. 256k cmos sram with variable impedance data-line loads. *IEEE Journal of Solid State Circuits*, SC-20, 1985.

[29] T. Chappell et. al. A 2-ns cycle, 3.8-ns access 512-kb cmos ecl sram with a fully pipelined architecture. *IEEE Journal of Solid State Circuits*, 26(11):1577–1585, Nov. 1991.

[30] T. Kobayashi et. al. A current-controlled latch sense amplifier and a static power saving input buffer for low-power architecture. *IEEE Journal of Solid State Circuits*, 28(4):523–527, Apr. 1993.

[31] T. Mori et. al. A 1 v 0.9 mw at 100 mhz 2 k*16 b sram utilizing a half-swing pulsed-decoder and write-bus architecture in 0.25 $\mu mdual - v_t$ cmos. *IEEE International Solid State Circuits Conference, Digest of Technical Papers*, 1998.

[32] T. Uetake et. al. A 1.0 ns access 770 mhz 36 kb sram macro. *Symposium on VLSI circuits: Digest of tech. papers*, 1999.

[33] T.N. Blalock et. al. A high-speed clamped bit-line currentmode sense amplifier. *IEEE JSSC*, 1991.

[34] Toshihiko Hirose et. al. A 20-ns 4-mb cmos sram with hierarchical word decoding architecture. *IEEE International Solid State Circuits Conference,Digest of Technical Papers*, 1990.

[35] W. Lee et. al. A 1v dsp for wireless communications. *IEEE International Solid State Circuits Conference Digest of Technical Papers*, 1997.

[36] R. Geiger, N. Strader, and P. Allen. Vlsi design techniques for analog and digital circuits. *McGraw-Hill*, 1990.

[37] R. Heald and J. Holst. A 6-ns cycle 256kb cache memory and memory management unit. *IEEE Journal of Solid State Circuits*, 28, Nov. 1993.

[38] C. Hwang. Semiconductor memories in it era. *IEEE International Solid State Circuits Conference, Digest of Technical Papers*, 2002.

[39] R. C. Jaeger. Comments on an optimized output stage for mos integrated circuits,. *IEEE Journal of Solid State Circuits*, SC-10(3):185–186, June 1975.

[40] J. Lachman and J. Micheal Hill. A 500mhz 1.5mb cache with on-chip cpu. *IEEE Solid-State Circuits Conference Digest of Technical Papers*, Feb. 1999.

[41] C. Mead and L. Conway. Introduction to vlsi systems. *Reading, MA, Addison-Wesley*, 1980.

[42] Aiyappan Natarajan and Wayne Burleson. A study of sense amplifiers for advanced microprocessor caches in 70nm technology.

[43] Nobutaro Shibata. Current sense amplifiers for low-voltage memories. *IEICE Trans. Electron.*, E79-C(8):1120–1130, Aug. 1996.

[44] I. E. Sutherland and R. F. Sproull. Logical effort: Designing for speed on the back of an envelope. *Advanced research in VLSI*, 1991.

[45] A Tang and C. Toumazou. High performance cmos current comparator. *Electronic Letters*, 30(1):5–6, 1994.

[46] B. Wicht. Current sense amplifier for embedded sram in high-performance soi designs. *Advanced Microelectronics*, 2003.